

١٩٩٠

سلسلة ملخصات شوم
نظريات ومسائل فن

البرمجة بالفورتران

يتضمن الفورتران الهيكل

مورتيتر

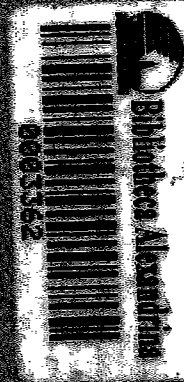
أرشبو

يحتوي الكتاب على ٣٧٥ مسألة محلولة

سلسلة ملخصات شوم في الحاسبات

الدار العالمية للنشر والتوزيع

القاهرة - الكويت - لندن



ملخصات شوم
نظريات ومسائل
في

البرمجة بالفورتران

يتضمن الفورتران الهيكلي
٠١١٨٧

تأليف

Ph. D. آرثر بو
أستاذ مشارك علوم الحاسب
جامعة تمبل

Ph. D. سيمور ليشتنز
أستاذ الرياضيات
جامعة تمبل

ترجمة

ابنسام صديق أبو الخير
ماجستير حسابات علمية
نايب مدير مركز بحوث الحسابات العلمية
والإحصائية - جامعة القاهرة
جمهورية مصر العربية

ماجدة صلاح الدين سلامة
مخطط برامج - مركز بحوث الحسابات
العلمية والإحصائية - جامعة القاهرة
جمهورية مصر العربية

مراجعة

الأستاذ الدكتور أحمد عزيز كمال
أستاذ بكلية الهندسة - جامعة القاهرة
جمهورية مصر العربية



الدار الدولية للنشر والتوزيع

القاهرة - الكويت - لندن

حقوق النشر

الطبعة الانجليزية : حقوق التأليف C ١٩٧٨ دار ماكجروهيل للنشر، انك . جميع الحقوق محفوظة .

Programming with Fortran
Including Structured Fortran
Seymour Lipschutz
Arthur Poe

الطبعة العربية الأولى : حقوق الطبع والنشر C ١٩٨٤ ، دار ماكجروهيل للنشر ، جميع الحقوق محفوظة .

الطبعة العربية الثانية : حقوق الطبع والنشر C ١٩٨٩ الدار الدولية للنشر والتوزيع ، جميع الحقوق محفوظة .

الطبعة العربية الثالثة : حقوق الطبع والنشر C ١٩٩٠ ، جميع الحقوق محفوظة للناشر .

الدار الدولية للنشر والتوزيع

٢٨ ش الأمرام - روكسى - مصر الجديدة

ص . ب : ٥٥٩٩ هليوبوليس غرب - القاهرة

ت : ٢٥٨٢٨٨٧ - تليكس PBCRB ٢٠٠٧٠

فاكس . ٢٠٢ / ٢٩١٨٠٥٩ ..

لا يجوز نشر أى جزء من هذا الكتاب أو اختزان مادته بطريقة الاسترجاع أو نقله على أى نحو أو
بأى طريقة ، سواء كانت اليكترونية أو ميكانيكية أو بالتصوير أو بالتسجيل أو خلاف ذلك إلا بموافقة
الناشر على هذا كتابة ومقديماً .

ISBN 07 084821

بسم الله الرحمن الرحيم

مقدمة الناشر

المعرفة هي أصل الحضارة ،
والكلمة هي مصدر المعرفة ،
والكلمة المطبوعة هي أهم مكون في هذا المصدر .

وقد كانت الكلمة المطبوعة ولاتزال أهم وسائل الثقافة والاعلام وأوسعها انتشاراً وأبقاها أثراً ،
حيث حملت إلينا حضارات الأمم عبر آلاف السنين لتتولى الأجيال المتلاحقة صياغة حضاراتها وإضاءة
الطريق بنور العلم والمعرفة .

والكلمة تبقى مجرد فكرة لدى صاحبها حتى تتاح لها فرصة نشرها وترجمتها إلى لغات الآخرين ثم
توزيعها ، وذلك وحده هو الذي يكفل لها أداء رسالتها .

وعالم الكتب العلمية عالم رحب ممتد الآفاق ، متسع الجنبات ، والعلم لا وطن له ولا حدود ، ويرى
يحتل القارئ بأحدث الكتب العلمية باللغة العربية لهو اليوم الذي تتطلع له الأمة العربية جمعاء .

والدار الدولية للنشر والتوزيع تشعر بالرضا عن مساهمتها في هذا المجال بتقديم الطبعة العربية
للكتب العلمية الصادرة عن دار ماكجروهيل للنشر بموجب الاتفاق المبرم معها ، مستهدفة توفير
احتياجات القارئ العربي أستاذاً وباحثاً وممارساً .

ومن جانب آخر فنحن نمد يدنا إلى الجامعات العربية والمراكز العلمية والمؤسسات والهيئات الثقافية
للتعاون معنا في إصدار طبعة عربية حديثة من الكتب والمراجع العلمية تخدم التقدم العلمي والحضارى
للقارئ العربي .

والله ولي التوفيق

محمد وفائى كامل
مدير عام
الدار الدولية للنشر والتوزيع

تقديم

يمكن أن يطلق على النصف الثاني من القرن العشرين عهد الحاسبات . فالناس في جميع مجالات العمل تقريباً (مثل الهندسة ، العلوم الطبيعية ، الاقتصاد ، علم النفس ، التعليم ، العلوم الاجتماعية ، العلوم الطبية ، القانون والأعمال ، أى في كل مجال يحتاج لجمع البيانات وتحليلها بواسطة الحاسبات سيكون لهم اتصال ما بالحاسبات واللغات المستعملة في توجيهاها . وتسمى إحدى هذه اللغات فورتران FORTRAN وهي اختصار FOR mula TRANslation . (أى ترجمة المصغير الرياضية) . وقد تم تصميم وكتابة هذا الكتاب لتقديم لغة الفورتران ولتعليم طرق حل المسائل باستخدام هذه اللغة . والهدف الأساسي لهذا الكتاب هو تعليم القارئ كيف يكتب برامج فورتران واضحة وفعالة بواسطة التركيز على الأساليب الفنية والخبرات الجيدة للبرمجة وذلك بالإضافة إلى تقديم قواعد الفورتران . يناقش هذا الكتاب كل المبادئ الهامة للفورتران القياسي وكذلك بعض السمات ككتاب دراسي لمقرر تمريني للبرمجة بالفورتران ، أو ككامل للكتب الدراسية المعتادة في المادة الدراسية « مقدمة لعلم الحاسبات الآلية » .

سيروق هذا الكتاب لعدد كبير من القراء ، كما أنه سيكون بمثابة دليل فعال للتعليم الذاتي ، ويرجع ذلك لمنهجه المبسط وكذلك تدرج أمثله . يبدأ كل فصل بجملته واضحة عن التعاريف والأساسيات المتعلقة بالموضوع مع مادة توضيحية ووصفية . يلي ذلك مجموعة متدرجة من المسائل المحلولة والتكيفية . فالمسائل المحلولة تستخدم في توضيح وتقوية المادة بينما تقدم المسائل التكيفية مراجعة كاملة للمادة المقدمة في الفصل .

ينقسم الكتاب إلى اثني عشر فصلاً . يتناول الفصل الأول باختصار العمليات الأساسية للحاسب . وذلك من خلال استخدام برامج بسيطة وهذا يعطى القارئ لمحة عن لغة الفورتران وكذلك بعض الشهور بالدينامية المتضمنة في البرمجة بالفورتران . ويناقش أيضاً هذا الفصل تعليمات تثقيب وتنظيم حزمة الفورتران حتى يتمكن القارئ من كتابة وتشغيل بعض البرامج البسيطة من بداية تعلمه للغة . يبدأ التقديم الرسمي للغة الفورتران ، في الفصل الثاني ، وفيه تقدم التعبيرات الرياضية وجمل التخصيص الحسابية وتناقش أيضاً حسابات الأعداد الصحيحة والحقيقية ، وكذلك بعض المثرات التي تتعرض لها الحسابات بالحاسب . يتعلق الفصل الثالث بعمليات الإدخال / الإخراج العددية . وكذلك يناقش كل من الملامح المصاغة وغير المصاغة .

يقدم الفصل الرابع طريقة رسم خرائط سير العمليات وذلك لمساعدة القارئ في وضع وتصور خطوات حل المسائل . وأهم من ذلك ، فالفصل الرابع يناقش جمل نقل التحكم في الفورتران الأساسي (غير الهيكل) بما في ذلك جملة IF المنطقية . ويناقش الفصل الثاني عشر ملامح التحكم الهيكل ، بما في ذلك كتابة جملة IF . بالمثل يقدم الفصل الخامس حلقة DO الأساسية المفهرسة . بينما يتم معالجة حلقة DO المعقدة وهياكل FOR و WHILE في الفصل الثاني عشر . ويناقش الفصل السادس المجموعات المتراسة والمنتخبات ذات الأدلة . ولمساعدة القارئ على كتابة برامج معقدة في وحدات قائمة بذاتها ، يقدم الفصل السابع الدوال FUNCTIONS والبرامج الفرعية SUBROUTINE .

يمالج الفصل الثامن أساليب البرمجة للبحث والفرز وصيانة الملفات الخ . كما تناقش الحسابات العددية مثل إيجاد أصفار الدوال وحل نظم المعادلات الخطية . وتعالج معلومات الحروف والمنتخبات المنطقية مع الإدخال / الإخراج في الفصل التاسع . يغطي الفصل العاشر ملامح إضافية من الإدخال / الإخراج مثل حقل G - وصيغة وقت التنفيذ . يقدم الفصل الحادي عشر ملامح أخرى للفورتران

مثل متغيرات DOUBLE PRECISION و COMPLEX وجمل COMMON و EQUIVALENCE . وكما سبق أن ذكرنا ، فالفصل الثاني عشر يناقش ملامح الفورتران الميكلي مثل كتابة جملة IF وحلقة DO المممة . وهذا الفصل متوافق مع المتغيرات الحديثة في الفورتران القياسي ومع مترجمات WATFOR و WATFIV .

كتبت المواضيع والفصول بطريقة تجعلها تكاد تكون مستقلة بعضها عن بعض . وذلك بنرض جعل الكتاب أكثر مرونة ويستفاد منه كرجع . وقد ضم أيضاً ملحفاً للتمثيل الداخلي للبيانات في الحاسب .

نود أن نشكر الكثيرين من الأصدقاء والزملاء على المقترحات القيمة والمراجعة الدقيقة لأصول الكتاب . نود أيضاً أن نعبر عن امتناننا لأعضاء هيئة ماكجروهيل لسلسلة تشوم وعلى الأخص جون اليانو والين لابريرا لماونتهما الدائبة .

سيمور ليبشتز

أرثر ت . بو

المحتويات

الصفحة

٧	المقدمة وتنظيم البرنامج
١-١	١-١ مقدمة
١-١	٢-١ تثقيب جمل الفورتران
١-١	٣-١ حزمة الفورتران
١-١	٤-١ تخزين الأعداد
١-١	٥-١ قراءة البيانات
١-١	٦-١ اتخاذ القرارات
١-١	٧-١ نظرة عامة على الحاسبات واللغات

٢٦	جمل رياضية
١-٢	١-٢ مقدمة
٢-٢	٢-٢ ثوابت عديدة (الأعداد)
٢-٢	٣-٢ أسماء المتغيرات (أسماء أماكن التخزين)
٢-٢	٤-٢ جمل النوع - صحيح وحقيقي
٢-٢	٥-٢ عمليات حسابية - حسابات صحيحة وحقيقية
٢-٢	٦-٢ التعميرات الحسابية
٢-٢	٧-٢ العمليات الحسابية ذات النمط المختلط
٢-٢	٨-٢ الدوال الرياضية
٢-٢	٩-٢ جملة التخصيص الحسابية
٢-٢	١٠-٢ رياضيات الحاسب .

٥١	الإدخال / الإخراج العدي
١-٣	١-٣ مقدمة
٢-٣	٢-٣ إدخال / إخراج غير مصاغ
٣-٣	٣-٣ مقدمة للإدخال / الإخراج المصاغ
٣-٣	٤-٣ الإدخال المصاغ ومواصفات جمل الإدخال
٣-٣	٥-٣ جمل WRITE المصاغ وتحكم العربة
٣-٣	٦-٣ مواصفات جمل الإخراج
٣-٣	٧-٣ الحقل الحرفي
٣-٣	٨-٣ السجلات ، السجلات المتعددة ، الشرط المائلة (/)
٣-٣	٩-٣ معامل التكرار
٣-٣	١٠-٣ برنامج بسيط كامل .

٩٣	نقل التحكم ، محركات سبر العمليات
١-٤	١-٤ مقدمة
٢-٤	٢-٤ الانتقال غير المشروط
٤-٤	٣-٤ الانتقال المشروط
٤-٤	٤-٤ جمل IF المنطقية
٤-٤	٥-٤ جمل IF الحسابية
٤-٤	٦-٤ التحكم في الحلقة التكرارية
٤-٤	٧-٤ جمل GOTO المحسوبة
٤-٤	٨-٤ جمل GOTO المحسوبة
٤-٤	٩-٤ الخوارزميات
٤-٤	١٠-٤ بطاقة المقدمة والبطاقة الخلفية .

١٣٥	حلقات DO التكرارية
١-٥	١-٥ مقدمة
٥-٥	٢-٥ جمل CONTINUE
٥-٥	٣-٥ استخدامات بسيطة لجملة DO
٥-٥	٤-٥ جمل DO
٥-٥	٥-٥ قوانين على استخدام حلقة DO التكرارية
٥-٥	٦-٥ الخروج من حلقة DO التكرارية
٥-٥	٧-٥ الانتقال بداخل وإلى حلقة DO تكرارية
٥-٥	٨-٥ ضرورة جملة CONTINUE
٥-٥	٩-٥ حلقات DO التكرارية المتداخلة

١٦٤	المجموعات المترابطة والمتغيرات ذات الأدلة
١-٦	١-٦ مقدمة
٦-٦	٢-٦ مجموعات مترابطة ذات بعد واحد
٦-٦	٣-٦ جمل DIMENSION
٦-٦	٤-٦ التعميرات الحسابية للأدلة
٦-٦	٥-٦ أمثلة لاستخدام مجموعات مترابطة
٦-٦	٦-٦ المجموعات المترابطة ذات الأبعاد المتعددة
٦-٦	٧-٦ إدخال إخراج مجموعة مترابطة وحلقات DO الضمنية
٦-٦	٨-٦ مثال لأسلوب البرمجة .

الصفحة

الفصل السابع : الدوال والبرامج الصغيرة الفرعية
 ٢٠٠ ١ - ٧ مقدمة ٢ - ٧ الدوال كبرامج فرعية ٣ - ٧ استدعاء البرامج الفرعية FUNCTION
 ٤ - ٧ دوال البرامج الفرعية FUNCTION الحاسبة لعدة قيم ٥ - ٧ مجموعات مترابطة وبرامج فرعية
 FUNCTION ، أبعاد متغيرة ٦ - ٧ دوال الجملة الحاسوبية ٧ - ٧ SUBROUTINES
 برامج صغيرة فرعية ٨ - ٧ مقارنة SUBROUTINE مع FUNCTION .

الفصل الثامن : أساليب البرمجة والحسابات العددية
 ٢٢٥ ١ - ٨ مقدمة ٢ - ٨ الفرز ٣ - ٨ الإدماج ٤ - ٨ البحث ٥ - ٨ التحديث
 ٦ - ٨ طريقة هورنر ٧ - ٨ حل معادلات معينة ٨ - ٨ التكامل العددي ٩ - ٨ المتجهات
 والمصفوفات ١٠ - ٨ المعادلات الخطية .

الفصل التاسع : معلومات الحروف ، المتغيرات المنطقية والعمليات الحسابية
 ٢٥٣ ١ - ٩ مقدمة ٢ - ٩ تخزين الحروف ٣ - ٩ حقل A ٤ - ٩ مناولة المعلومات الحرفية
 ٥ - ٩ حقل H ٦ - ٩ ثوابت منطقية ومتغيرات منطقية ٧ - ٩ المعاملات والتعبيرات المنطقية
 ٨ - ٩ تخصيص القيم المنطقية ٩ - ٩ حقل L ١٠ - ٩ التدرج المرص للعمليات الحسابية .

الفصل العاشر : ملاحق إضافية للإدخال / الإخراج
 ٢٧٨ ١ - ١٠ مقدمة ٢ - ١٠ DATA ٣ - ١٠ حقل T ٤ - ١٠ حقل G
 ٥ - ١٠ معامل التدرج ٦ - ١٠ قاعدة التوقاس اليسرى ٧ - ١٠ صيغ وقت التنفيذ
 ٨ - ١٠ الرسم البياني .

الفصل الحادي عشر : ملاحق متنوعة للفورتران
 ٣٠٦ ١ - ١١ مقدمة ٢ - ١١ النوع (TYPE) ٣ - ١١ جملة IMPLICIT ٤ - ١١ الدقة
 المتضاعفة ٥ - ١١ الأعداد المركبة ٦ - ١١ جملة GO to الخاصة وحل ASSIGN
 ٧ - ١١ المداخل والرجوع المتعدد لبرنامج فرعي ٨ - ١١ جملة COMMON غير المميزة
 ٩ - ١١ جملة COMMON المميزة ١٠ - ١١ جملة EQUIVALENCE ١١ - ١١ كتلة
 البيانات ١٢ - ١١ خارجي EXTERNAL .

الفصل الثاني عشر : الفورتران الهيكل
 ٣٢٤ ١ - ١٢ هيكل IF ٢ - ١٢ هيكل التحكم في الحلقة التكرارية .

ملحق (أ) : دوال المكتسبة
 ٣٥١
 ملحق (ب) : التمثيل الداخلي للبيانات
 ٣٥٢
 المصطلحات العلمية (عربي - إنجليزي)
 ٣٥٩
 المصطلحات العلمية (إنجليزي - عربي)
 ٣٦٦
 الفهرس الأبجدي
 ٣٧٢

الفصل الأول

المقدمة وتنظيم البرنامج

١ - ١ مقدمة

يمكن تسمية النصف الثاني من القرن العشرين بعصر الحاسبات الآلية ولقد انتشرت هذه الآلات و « اللغات » المتعددة المستخدمة في توجيهها حتى أصبحت تقارب انتشار الآلات الكاتبة والمسابر الحاسبة . يقوم الحاسب الآلي أساساً بتنفيذ ثلاث وظائف :

١ - استقبال بيانات (إدخال)

٢ - تشغيل البيانات بحسابات متعددة

٣ - إصدار بيانات (إخراج)

تم هذه الوظائف بإعطاء الحاسب برنامجاً يتكون من سلسلة من الجمل (تسمى مجموعة من الجمل المتتالية جزءاً من البرنامج) .

والهدف الأساسي من هذا الكتاب هو تعلم القارئ كيف يكتب برنامج للحاسب باستخدام لغة الفورتران . وكلمة فورتران FORTRAN مأخوذة من FORMula TRANslation أى ترجمة الصيغ الرياضية) وستبدأ هذه الدراسة رسمياً من الفصل التالي . سيغطي هذا الفصل بعض الاعتبارات الخاصة بتثقيب جمل الفورتران على البطاقات وكيفية إعداد حزمة الفورتران للحاسب . وسنتعلم أيضاً بعض الحقائق عن الحاسبات والتي يمكن أن تعطينا إحساساً بالدينامية المتضمنة في البرمجة بلغة الفورتران . وسنرى بالتحديد كيف يقوم الحاسب بتنفيذ بعض البرامج البسيطة . (وإنما لنجد أن يقوم القسارىء فعلاً بتثقيب برنامج أو أكثر من هذه البرامج على بطاقات وتشغيلها على الحاسب) .

وسنختم الفصل بنظرة شاملة على الحاسبات ولغاتها . وتشتمل هذه النظرة على تفاصيل كثيرة لن نحتاجها على الإطلاق في كثير من المواضيع اللاحقة في هذا الكتاب . ولذلك يمكن أن يمر القارئ على هذه الأجزاء مروراً سريعاً في أول قراءة لهذا الكتاب وأن يرجع إليها عند الضرورة .

١ - ٢ تثقيب جمل الفورتران

أحياناً نبدأ بكتابة برامج الفورتران على صحائف تكويد (أنظر شكل ١ - ١) . إلا أنه بمجرد كتابتها يجب إعطاؤها للحاسب بشكل يمكن للآلة أن تقرأه مثل مجموعة بطاقات مثقبة أو شريط ورق أو شريط مغنط ، أو من خلال آلة كاتبة طرفية وذلك يتوقف على وحدة الإدخال . وسنفتراض خلال هذا الكتاب أن وحدة الإدخال الخاصة بنا هي وحدة قراءة البطاقات المثقبة ، وعلى ذلك فسوف تثقب البرامج على بطاقات . تحتوي هذه البطاقة على 80 عموداً كما في الشكل ١ - ٢ (أ) . بحيث يمكن أن يتسع كل عمود لحرف واحد .

يجب أن تثقب كل جملة في برنامج الفورتران على بطاقة منفصلة . وكما هو موضح في شكل ١ - ٢ (ب) ، ليس من الضروري أن تستخدم جملة الفورتران جميع الأعمدة الثمانية بالبطاقة . بالتحديد لدينا القوانين التالية :

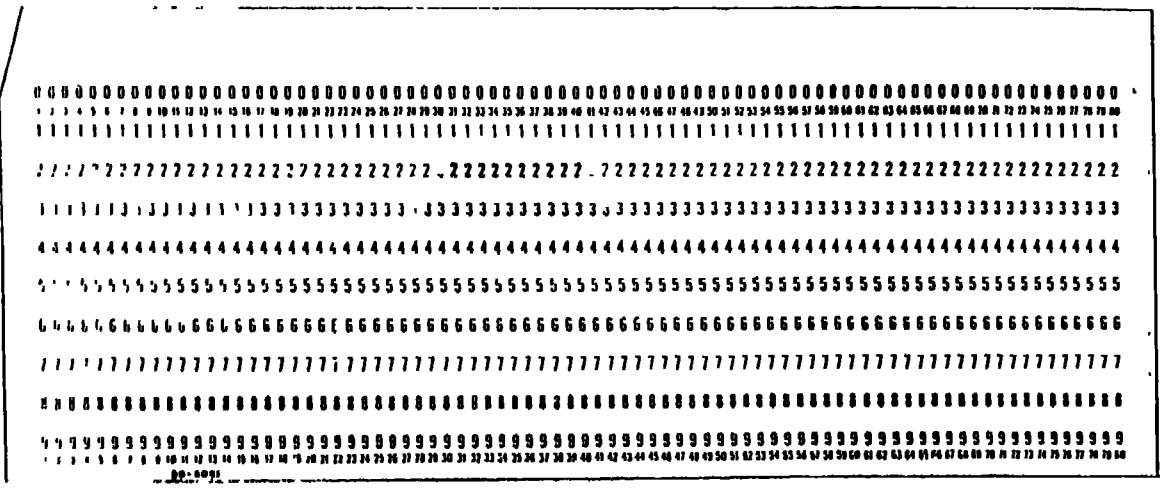
١ - تثقب جمل الفورتران الأصلية من العمود 7 إلى العمود 72 فقط (شاملاً 7 و 72)

الفصل الأول : المقدمات وتنظيم البرنامج

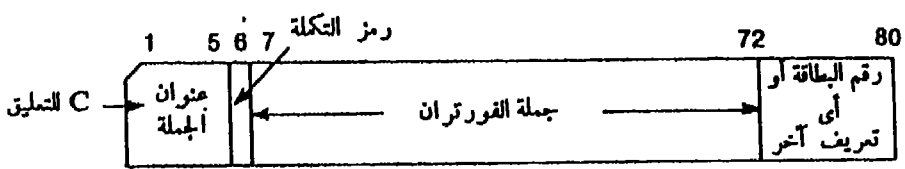
٨

STATEMENT Number	FORTRAN STATEMENT	IDENT
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		
72		
73		
74		
75		
76		
77		
78		
79		
80		

شكل ١ - ١ صحيفة تكرود الفورتران



(١)



(ب)

شكل ٢ - ١

الفصل الأول : المقدمات وتنظيم البرنامج

٩

١ - يسمى عمود 6 عمود التكلفة ويدخر للفرض التالي : إذا ما كانت جملة الفورتران أطول من أعمدة البطاقة وعددها 66 كما ذكرنا سابقا ، فيمكن تكلفتها حينئذ في بطاقات أخرى (بين الأعمدة 7 إلى 72) بما لا يزيد عن 19 بطاقة إضافية . في هذه الحالة يجب أن يثقب « 1 » (أو أى علامة أخرى غير الصفر) في العمود 6 من كل بطاقة من البطاقات الإضافية . بمعنى أن أى علامة (غير الصفر) مثقبة في العمود 6 تدل الحاسب أن هذه الجملة هي تكلفة للجملة السابقة .

٢ - يمكن وضع عناوين للاستدلال عليها ، وهذه العناوين تثقب من العمود الأول إلى العمود الخامس من بطاقة الجملة المعنونة . والعنوان في الفورتر - هو أى رقم صحيح موجب بدون إشارة جبرية (بحد أقصى خمس خانات) ويسمى رقم الجملة . ومن الواضح أنه لا يجوز أن يتكرر رقم الجملة لجلتين مختلفتين .

٤ - يتجاهل الحاسب محتويات الأعمدة ابتداء من العمود 73 إلى 80 . ومن الممكن تثقيب هذه الأعمدة للتعريف أو للتسلسل أو لأى غرض آخر .

يمكن إضافة تعليقات إلى البرنامج بتثقيب « C » في العمود الأول من البطاقة . ويثقب التعليق نفسه في أى مكان بين العمود 2 و العمود 80 . ولتمثل التعليقات أى عبء حقيق على البرنامج . فهي تستخدم عادة لتعريف البرنامج وللإيضاح ، وببساطة يتخلى الحاسب التعليق ويتحرك إلى الجملة التالية إلا أن التعليق سوف يظهر في صفحة الطباعة حين يقوم الحاسب بدباعتها .

ملحوظة : سوف يلاحظ القارئ عند تثقيب البطاقات أن الحروف الكبيرة فقط هي المتاحة بالنسبة للحروف الإيجدية . تكتب علامات الجمع ، الطرح ، الضرب والقسمة كالأتي + ، - ، * ، / على الترتيب . أما * فترمز إلى الأس ، أى رفع الرقم إلى قوة ما (هذه العمليات ستناقش بالتفصيل في الفصل الثانى) وعلى هذا في الفورتران :

$$a \cdot b \text{ تكتب } A * B \text{ و } \frac{a}{b} \text{ تكتب } A / B \text{ و } a^b \text{ تكتب } A ** B$$

ونلاحظ أن الحاسب في الفورتران يتجاهل المسافات الخالية وعلى ذلك يمكن للقارئ، إضافتها لتسهيل عملية القراءة . فعمل سبيل المثال يمكن تثقيب

$$A ** 2 + B ** 2 \text{ بدلا مما يكافئه في } A ** 2 + B ** 2$$

$$4 * I + 6 * J \text{ بدلا مما يكافئه في } 4 * I + 6 * J$$

و GO TO بدلا من GOTO

١ - ٣ حزمة الفورتران

برنامج الفورتران الذى نكتبه ونثقبه على بطاقات هو مجرد جزء واحد من الحزمة التى نعطيها للحاسب إذا أريد تنفيذ البرنامج . وتتكون هذه الحزمة من سلسلة البطاقات التالية :

١ - بطاقة تعريف تحتوى هذه البطاقة عادة على اسم مخطط البرنامج ، المشروع ، المادة الدراسية أو رقم الحاسب ، ورقم الأولوية الحد الأقصى لوقت التشغيل ، الخ .

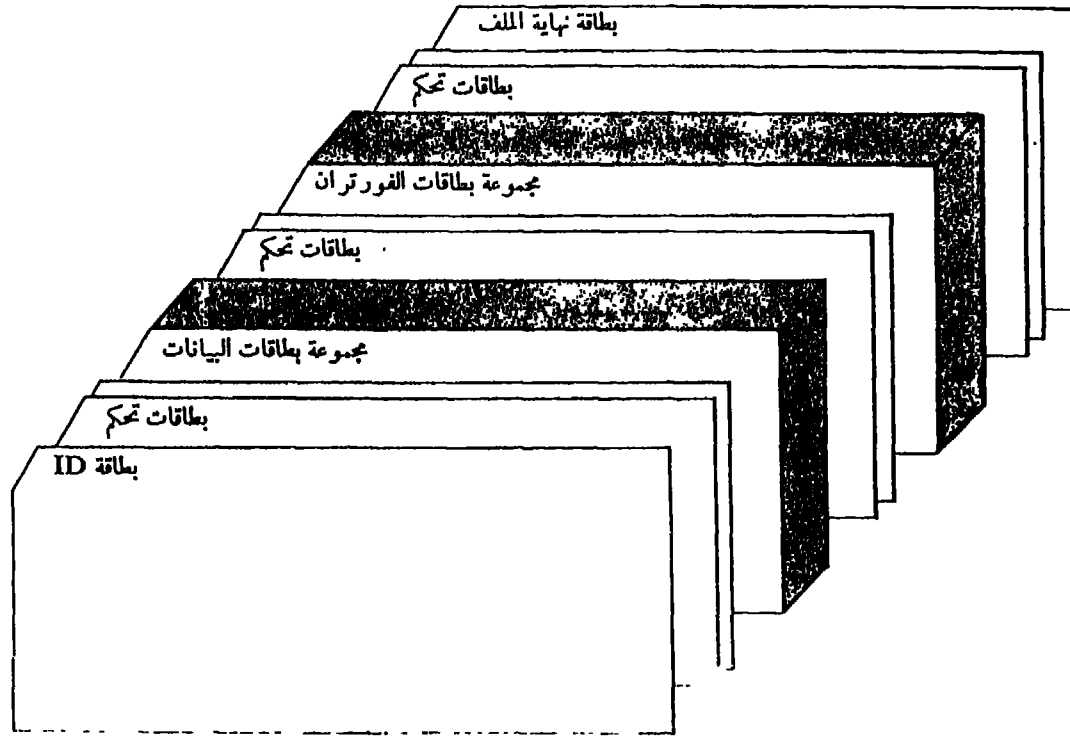
٢ - مجموعة بطاقات الفورتران ، تحتوى هذه المجموعة من البطاقات على برنامج الفورتران الفعل .

٣ - مجموعة بطاقات البيانات ، تحتوى هذه المجموعة من البطاقات ، والتي قد تكون خالية ، على البيانات التى تقرأ بواسطة الحاسب .

٤ - بطاقة نهاية الملف ، تحدد هذه البطاقة نهاية الحزمة .

من الممكن فصل المجموعات الأربع السابقة بواسطة بطاقة أو أكثر من بطاقات التحكم كما هو موضح في الشكل ١ - ٣ . والمعلومات الفعلية التي تحتويها بطاقة التعريف وبطاقة نهاية الملف وبطاقة التحكم تعتمد على الحاسب المستخدم .

تحتوي مجموعة بطاقات البيانات على المدخلات ، من الممكن استخدام كل أعمدة البطاقة الثمانية عند تثقيب بطاقات البيانات ، أى أن القوائم الخاصة بتثقيب جمل الفورتران لا تطبق في هذه الحالة . وليس لأى عمود (العمود 6 مثلا) ، أى غرض خاص .



الشكل ١ - ٣ حزمة الفورتران

١ - ٤ تخزين الأعداد

تتكون وحدة ذاكرة الحاسب من عدة خلايا ذاكرة (أماكن) لتخزين البيانات . ولايخصص مكان في الذاكرة لعدد إلا إذا أُعطي اسم لهذا المكان . ونمثل تخصيص عدد ، مثل 75 لمكان في الذاكرة يسمى N بواسطة

$$N \leftarrow 75$$

وبصورة عامة :

$$NAME \leftarrow \text{تعبير حسابي}$$

يدل على أن قيمة التعبير الحسابي يعطى لمكان الذاكرة المسمى NAME .

ونؤكد أن السهم \leftarrow مقبول عموماً للدلالة على « التخصيص » . إلا أن لغات البرمجة المختلفة تمثل \leftarrow بطرق مختلفة . ففي الفورتران يمثل السهم = (أى علامة التساوى الحسابية) . وعلى ذلك فإن جملة الفورتران :

$$I = 7$$

تأمر الحاسب بتحديد مكان الذاكرة المسمى I بالقيمة 7 . وجملة الفورتران :

$$J = 4 + 5 + 8$$

$$J = 4 + 5 + 8$$

تمطى التعليمات للحاسب ليقوم :

١ - بجمع القيم الموجودة على الجانب الأيمن من علامة = .

٢ - تخزين 'ح' ، وهو 17 ، في المكان المسمى J .

لاحظ تضمين إجرائين : إيجاد قيمة التعبير الرياضي على يمين العلامة = وتخزين الناتج في المكان المسمى على يسار العلامة = .

يمكن أن يحتوي التعبير الرياضي أيضاً على أسماء الأماكن تخزين إلا أن هذه الأسماء يجب أن تعرف بمعنى أنها أسماء الأماكن تخزين في الذاكرة تم فيها تخزين أرقام مسبقاً في نفس البرنامج . يوجد الحاسب قيمة التعبير الرياضي باستبدال الأسماء بالقيم المخزنة في أماكن التخزين هذه .

وعلى سبيل المثال أنظر جملة الفورتران :

$$K = 3 * I + 2 * J - 24$$

حيث تحتوي I و J على 7 ، 17 على الترتيب (تذكر أن النجمة * تعني الضرب في الفورتران) . يوجد الحاسب قيمة التعبير الرياضي على اليمين كالتالي :

$$3 * 7 + 2 * 17 - 24$$

ويخزن الناتج ، 31 ، في مكان الذاكرة المسمى K . لاحظ أن محتويات I و J لا تتحرك من الأماكن I و J حين يحسب الحاسب قيمة $24 - 2 * J - 3 * I$ بل تنسخ القيم في الوحدة الحاسوبية ، وتحفظ I و J بنفس قيمها .

وسنوضح هذه الأفكار في برنامجنا الأول .

البرنامج الأول

نكتب برنامجاً لحساب المجموع LS وحاصل الضرب LP للأرقام $I = 24$ ، $J = 3$ ، $K = 7$ ، شكل ١ - ١ (أ) هو صحيفة التكويد لبرنامج الفورتران الذي يقوم بهذه الحسابات ، وشكل ١ - ٢ (ب) يعطى صورة البرنامج بعد تثقيبه على مجموعة من البطاقات .

لاحظ أن كل جملة مكتوبة على سطر منفصل في صحيفة التكويد ومثقة على بطاقة منفصلة . وسنتابع الحاسب أثناء قراءته وتنفيذه للبرنامج وإليك أولاً بعض التفسيرات لمدة جمل في البرنامج :

١ - أول بطاقة تحتوي على الحرف « C » مثقبة في العمود الأول ، ولذلك فهي بطاقة تعليق . كما تم توضيح ذلك من قبل بالتعليقات لا تؤثر على البرنامج ، فمتدا يقابل الحاسب الحرف « C » في العمود الأول فإنه ببساطة يقوم بطباعة التعليق ويمضي إلى الجملة التالية .

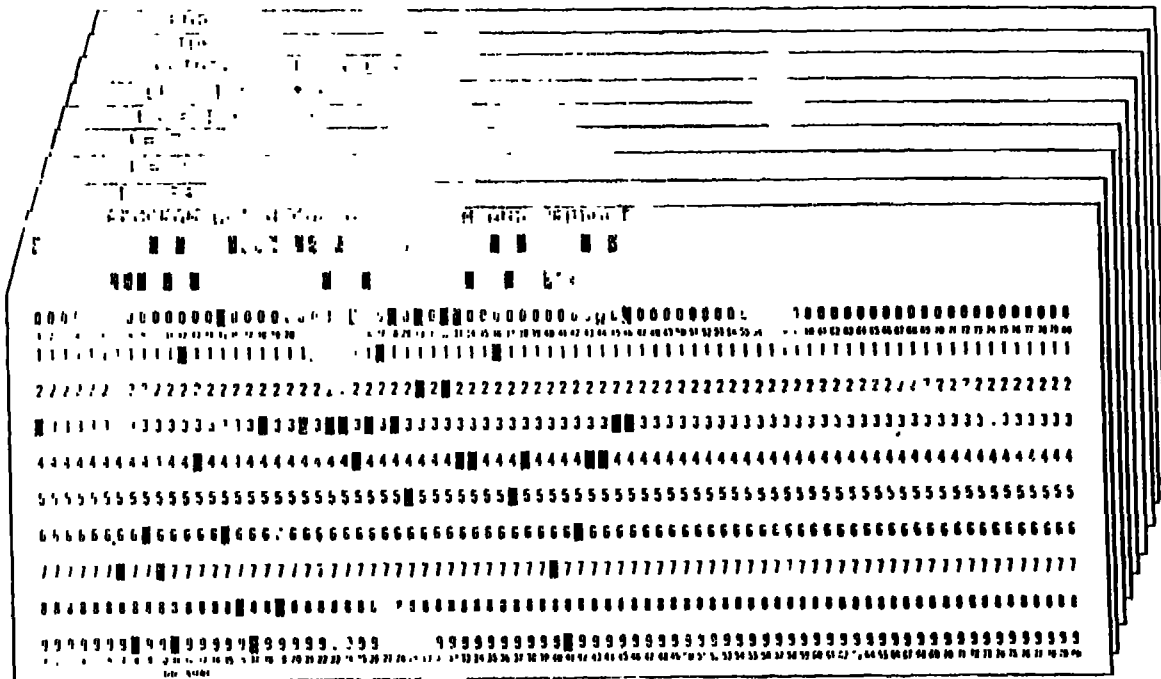
٢ - جملة PRINT

PRINT, I, J, K, LS, LP

تأمر الحاسب بطباعة المحتويات الحالية لأماكن الذاكرة I و J و K و LS و LP بهذا الترتيب . هذه جملة إخراج غير مصاغة . وتطبع القيم في صيغة محددة مسبقاً وسندرس بالتفصيل الجمل المصاغة وغير المصاغة في الفصل الثالث .

"C" FOR COMMENT		FORTRAN STATEMENT				
STATEMENT NUMBER	5	10	20	30	40	50
C.		PROGRAM WHICH CALCULATES SUM AND PRODUCT.				
		I = 12.4				
		J = 3				
		K = 7				
		LS = I+J+K				
		LP = I*J*K				
		PRINT, I, J, K, LS, LP				
		STOP				
		END				

(أ)



(ب)

شكل ١ - ٤

جملة STOP تأمر الحاسب بإيقاف التنفيذ ، وجملة END هي دائماً آخر جملة في أى برنامج . (أنظر قسم ٨ - ١ .) (مزيد من التفسير عن الفرق بينهما) .

تنفيذ البرنامج

السطر الأول : هذا تعليق لوجود « C » مطبوعة في العمود الأول . وكما سبق ذكره فالحاسب يتخطى هذه الجملة ويمضى إلى الجملة التالية .
السطر الثاني : يأمر الحاسب بتخزين الرقم المح 24 في مكان الذاكرة المسمى I :

$$I \leftarrow 24$$

السطر الثالث : $J \leftarrow 3$

السطر الرابع : $K \rightarrow 7$

السطر الخامس : يأمر الحاسب بإيجاد قيمة التمييز الرياضى على يمين علامة التساوى بجمع $7 + 3 + 24$ ثم تخزين الناتج ، وهو 34 ، في مكان الذاكرة المسمى LS .

$$LS \leftarrow 34$$

السطر السادس : يأمر الحاسب بإيجاد قيمة التمييز الرياضى على اليمين ، بضرب 24 في 3 في 7 ، وتخزين الناتج في LP :

$$LP \leftarrow 504$$

السطر السابع : يأمر الحاسب بطباعة القيم المخزنة في I و J و K و LS و LP بهذا الترتيب ، ونمثل هذا بإسالة المخرج بمتوازي أضلاع كالتالى :

24, 3, 7, 34, 504

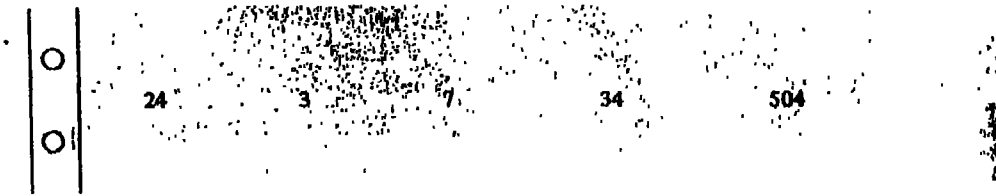
(انظر إلى التعليق في آخر هذا القسم على جملة PRINT غير المصاغة هذه) .

السطر الثامن : تأمر الحاسب بإيقاف التنفيذ .

السطر التاسع : يجب أن ينتهى كل برنامج بجملة END .

المخرج

سنعتبر أن وحدة الإخراج هي آلة الطباعة . وسنعتبر أن ورق الطباعة الخاص بآلة الطباعة يحتوى السطر منها على 132 عموداً (مسافات الطباعة) . وتستخدم عادة جملة الصياغة FORMAT لتوجيه الحاسب أن يضع المخرج على ورق الطباعة . في هذا القسم سنستخدم جملة PRINT غير المصاغة ، وعلى هذا فهناك صيغة محددة مسبقاً للمخرج فسيظهر المخرج على سطر واحد كما في شكل ١ - ٥ .



شكل ١ - ٥

تعليق : إذا كان القارئ سينفذ البرنامج على حاسب لا يقبل جملة PRINT غير المصاغة فيجب على القارئ أن يستبدل جملة PRINT (السطر السابع) بالجمليتين التاليتين :

```
WRITE(6, 31) I, J, K, LS, LP
31 FORMAT(5I12)
```

(وسوف يتم شرح هذا الثنائي WRITE-FORMAT في الفصل الثالث)

١ - ٥ قراءة البيانات

قام البرنامج السابق بتوليد بياناته خلال التنفيذ بتخصيص قيم I, J, K, صراحة . من الممكن أيضاً أن « يقرأ » الحاسب بطاقات البيانات والشرائط الممنظفة أو أى وحدة إدخال أخرى . وحيث أننا افترضنا استعمال وحدة قراءة البطاقات ، فإن الحاسب سيقراً البيانات من مجموعة بطاقات . (و كما هو موضح في شكل ١ - ٣ ، تظهر مجموعة بطاقات البيانات بعد مجموعة بطاقات برنامج الفورتران في حزمة الفورتران المعطاة للحاسب) .

وتعطي التعليمات للحاسب بقراءة البيانات من بطاقات البيانات باستخدام جملة READ . وتبدو جملة READ غير المصاغة على النحو التالي :

```
READ, I, J, K, M, N
```

لاحظ أن كلمة READ وكل متغير ، فيما عدا الأخير ، متبوعة بفصلة (.) هذه الجملة تعطي تعليمات للحاسب بتخزين القيم الخمس الأولى من مجموعة البيانات في الأماكن المسماة I ، J ، K ، M ، N على الترتيب

والميزة الأساسية لاستخدام جملة READ هي أن نجعل البرنامج بصورة أعم ، مستعمل عن بيانات الإدخال . على سبيل المثال ، لو أننا استخدمنا جملة READ في البرنامج السابق لقراءة قيم للأماكن I ، J ، K ، لكان الممكن استخدام هذا البرنامج لإيجاد مجموع وحاصل ضرب أى قيم I ، J ، K ، وليس فقط للقيم ٣ ، ٤ ، ٧ .

مثال ١ - ١

(أ) سنعيد كتابة البرنامج السابق إلا أننا سوف نقرأ قيم I ، J ، K . بالتحديد سوف نستبدل ، الجمل الثلاث بعد التعليق بالجملة الواحدة .

```
READ, I, J, K
```

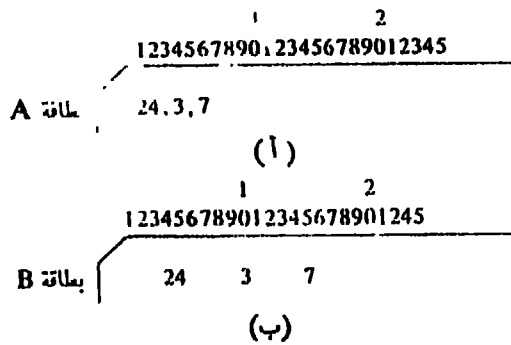
ويتقرب بطاقة مثل بطاقة A في الشكل ١ - ٦ (أ) . فإذا لم يكن متاحاً للقارئ استخدام إدخال وإخراج غير مصاغ فعليه استخدام الجمليتين :

```
READ(5, 10) I, J, K
```

```
10 FORMAT(3I5)
```

ويتقرب بطاقة بيانات بديلة للسابقة كما هو موضح في البطاقة B في الشكل ١ - ٦ (ب) . سيكون خرج هذا البرنامج هو نفس خرج البرنامج الأول .

(ب) إشرح الفرض من البرنامج



شكل ١ - ٦

```

READ, A, B, C, D, E
SUM = A + B + C + D + E
PRINT, A, B, C, D, E
PRINT, SUM
STOP
END

```

هذا البرنامج يحسب مجموع خمسة أرقام وتوحد هنا جملتنا PRINT . قيم A ، B ، C ، D ستطبع على سطر ويطبع مجموعها (SUM) على سطر آخر .

(٥) إشرح الغرض من البرنامج :

```

READ, B, H
AREA = (B*H)/2.0
PRINT, B, H, AREA
STOP
END

```

هذا البرنامج يحسب مساحة (AREA) المثلث ذو القاعدة B والإرتفاع H . وستطبع AREA, H, B على سطر واحد .

١ - ٦ اتخاذ القرارات

ينفذ الحاسب عادة جمل البرنامج واحدة تلو الأخرى كما في البرنامج السابق إلا أنه في بعض الأحيان نود أن يكرر أو يتخطى الحاسب جزءاً من البرنامج ، أي أننا نريد الحاسب أن ينتقل من نقطة في البرنامج إلى نقطة أخرى ويمكن تنفيذ ذلك باستخدام جملة GO TO والتي لها الشكل :

GO TO n

حيث n هو « رقم جملة » . فهذه الجملة تأمر الحاسب أن ينتقل التحكم إلى الجملة الممنونة n ، بمعنى أن الجملة التالية التي نريد تنفيذها هي الجملة التي رقمها n .

من أهم صفات الحاسب مقدرته على مقارنة القيم واتخاذ القرارات بناء على ذلك . على سبيل المثال ، أنظر جملة الفورتران :

IF(N.LT.300) GO TO 100

(الرمز " .LT." في الفورتران يقابل « أقل من » فجملة IF هذه تعطي للحاسب التعليمات كما يلي :

إذا تحقق الشرط $n < 300$ اذهب لتنفيذ الجملة التي رقمها 100 ، وأن لم يتحقق ذلك استكمل بتنفيذ الجملة التالية لجملة IF .

بمعنى آخر ، تسأل الجملة السؤال « IS N < 300? » . إذا كان الجواب على ذلك « نعم » ، ينتقل الحاسب التحكم إلى الجملة التي رقمها 100 ، أما إذا كان الجواب « لا » يستمر الحاسب في تنفيذ الجملة التالية .

يوضح البرنامج التالى هذه الأفكار (وستدرس هذه الأفكار بالتفصيل في الفصل الرابع) . ويعطى البرنامج للحاسب التعليمات بطباعة الأرقام فردية الصحيحة الموجبة من 1 إلى 11 وكذلك مربعاتها .

```

C   PROGRAM ODD INTEGERS AND THEIR SQUARES
    N = 1
21  K = N*N
    PRINT, N, K
    N = N + 2
    IF(N.LT.12) GO TO 21
    STOP
    END

```

قبل مناقشة تنفيذ البرنامج ، نلاحظ أن الجملة

$$N = N + 2$$

قد تبدو محيرة ، إلا أننا عرفنا من القسم ١ - ٤ أن علامة = في الفورتران لا تعنى التساوى ولكن تعنى تخصيص (والتي تتضمن إجرايين) أولاً يوجد الحاسب قيمة $N + 2$ باستعمال القيمة الحالية لـ N وبعد ذلك يخزن هذا الرقم الجديد في المكان N . بمعنى آخر أن الجملة $N = N + 2$ تعنى أن القيمة الجديدة لـ N هي القيمة الحالية مضافاً إليها 2 .

تنفيذ البرنامج

الخطوة :

١ - السطر الأول : عبارة عن تعليق حيث توجد C مثقبة في العمود الأول .

٢ - السطر الثاني : $N \leftarrow 1$

٣ - السطر الثالث : يأمر الحاسب بضرب قيمة N في نفسها ، وتخزين النتيجة في K :

$$K \leftarrow 1$$

٤ - السطر الرابع : تطبع قيم N و K



٥ - السطر الخامس : يأمر الحاسب بجمع 2 على القيمة الحالية لـ N ، وتخزين هذا المجموع (3) في المكان N (بهذه الطريقة تسمح القيمة السابقة لـ N) . وتمثل حقيقة مسح القيمة 1 الموجودة في N واستبدالها بالقيمة 3 بواسطة

$$N \leftarrow 3$$

٦ - السطر السادس : يجرى السؤال التالى :

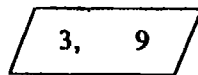
هل N أقل من 12 ؟

وحيث أن $N = 3$ ، فالجواب هو نعم ، أى أن التمييز الرياضى بين القوسين صحيح لذلك يذهب الحاسب إلى الجملة التى رقمها 21 ، أى إلى السطر الثالث من البرنامج .

٧ - السطر الثالث : يربع القيمة الحالية لـ N ويخزن الناتج (9) في المكان K (وعلى ذلك تسمح القيمة السابقة لـ K) .

$$K \leftarrow 9$$

٨ - السطر الرابع : يطبع القيم الحالية لـ N و K .



٩ - السطر الخامس : يجمع الحاسب 2 على القيمة الحالية لـ N ، وعلى ذلك تصيح

$$N \leftarrow 5$$

١٥ - السطر السادس : يعود الحاسب إلى الجملة التي رقبها 21 ، أي إلى السطر الثالث في البرنامج حيث أن الإجابة على السؤال « هل N أقل من 12 ؟ » هو نعم .

١١ - السطر الثالث : يربع القيمة الحالية لمتغيرات N ويخزن الناتج (وهو 25) في K

$$K \leftarrow X \times 25$$

١٢ - السطر الرابع : يطبع القيم الحالية لمتغيرات N و K .

$$\boxed{5, \quad 25}$$

١٣ - السطر الخامس : يزيد الحاسب قيمة N بمقدار 2

$$N \leftarrow X \times 2 + 7$$

١٤ - السطر السادس : يعود الحاسب إلى السطر الثالث حيث أن $7 < 12$.

١٥ - السطر الثالث : يربع الحاسب N ويخزن القيمة في K

$$K \leftarrow X \times 25 \times 49$$

١٦ - السطر الرابع : يطبع الحاسب قيم N و K .

$$\boxed{7, \quad 49}$$

١٧ - السطر الخامس : يزيد الحاسب قيمة N بمقدار 2 .

$$N \leftarrow X \times 2 + 9$$

١٨ - السطر السادس : يعود الحاسب إلى السطر الثالث حيث أن $9 < 12$.

١٩ - السطر الثالث : يربع الحاسب N ويخزن هذا العدد الصحيح في K .

$$K \leftarrow X \times 25 \times 49 \times 81$$

٢٠ - السطر الرابع : يطبع الحاسب قيم N و K .

$$\boxed{9, \quad 81}$$

٢١ - السطر الخامس : تزداد قيمة N بمقدار 2

$$N \leftarrow X \times 2 + 11$$

٢٢ - السطر السادس : يعود الحاسب إلى السطر الثالث حيث أن $11 < 12$.

٢٣ - السطر الثالث : تستبدل قيمة K الحالية بمربع N .

$$K \leftarrow X \times 25 \times 49 \times 81 \times 121$$

٢٤ - السطر الرابع : تطبع قيم N و K .

11, 121

٢٥ - السطر الخامس : تزداد قيمة N بمقدار 2

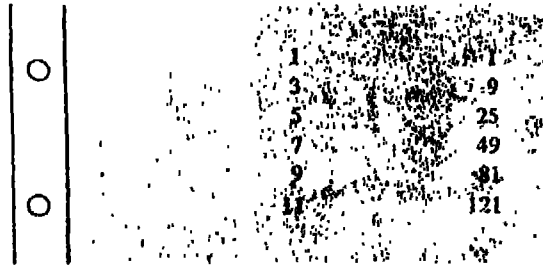
$N \leftarrow X \times X \times X \times X \times X \times X \times X \times X \times X \times X \times X \times X \times X \times X \times X \times X$

٢٦ - السطر السادس : ينفذ الحاسب الجمد التالية في البرنامج . حيث أن جواب السؤال هل $N < 12$ الآن هو لا ، أى أن التعبير الرياضى بين القوسين غير صحيح .

٢٧ - السطر الرابع : يأمر الحاسب بالتوقف STOP .

الخرج

في كل مرة تظهر جملة PRINT ، يطبع الحاسب المخرج في سطر واحد ومن ثم سيظهر كل عدد فردى ومرممه على سطر مستقل ، كما في الشكل ١ - ٧ .



شكل ١ - ٧

تعليق ١ : لو استبدلنا السطر السادس في البرنامج ، بما يلي مثلا :

IF(N.LT.100) GO TO 21

فإن الحاسب سيطلع الأعداد الصحيحة الفردية ومربعاتها من ١ إلى ٩٩ .

تعليق ٢ : إذا كان القارئ سينفذ برنامجاً على حاسب لا يقبل جملة PRINT غير المصاغة ، فعلى القارئ استبدال PRINT بالجملة التاليتين :

WRITE(6, 31) N, K
31 FORMAT(2I15)

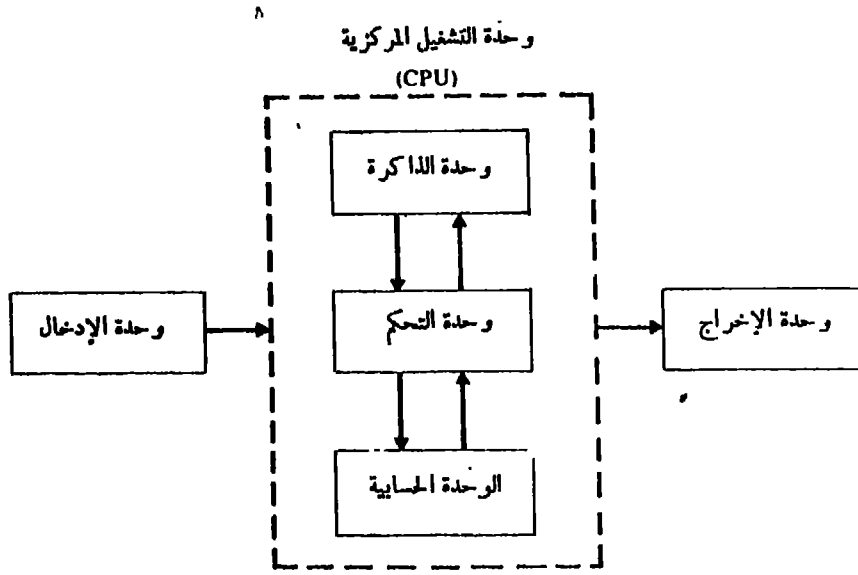
١ - ٧ نظرة عامة على الحاسبات واللفئات

يعطى هذا القسم نظرة عامة على الحاسبات وعلى لذات الحاسب . ويتم أن القارئ يمكن أن يمر على هذا القسم مروراً سريعاً في القراءة الأولى للكتاب ، إلا أنه توجد بعض مصطلحات وتفاصيل معينة ومطلوبة حتى يمكن تفهم المواضيع اللاحقة ، وعلى سبيل المثال ، حمل منفذة وغير منفذة . ولقد ضمنا هذا القسم كل الضروريات لكي يكون الكتاب كاملاً ، ويستطيع القارئ استخدام هذا القسم كمرجع عند الضرورة .

(أ) وحدات الحاسب

يتكون الحاسب من خمسة أجزاء أساسية : وحدة الإدخال ووحدة الإخراج ووحدة الذاكرة ووحدة التحكم والوحدة الحاسوبية . وهذه الوحدات موضحة في شكل ١ - ٨ .

تتكون وحدات الإدخال والإخراج من أجهزة تسمح للحاسب باستقبال أو عرض المعلومات . وتتكون في صورة وحدات لقراءة البطاقات أو شرائط مبنطة أو وحدات قراءة الشرائط الورقية أو وحدات الطباعة أو وحدات CRT (أنبوبية أشعة المهبط) طرفية أو وحدات التحكم الطرفية الكاتبة . بعض هذه الوحدات يمكن استخدامها للإدخال والإخراج معا .



شكل ١ - ٨

وتكون الوحدات الثلاث (وحدة الذاكرة ووحدة التحكم والوحدة الحسابية) مايسمى بوحدة التشغيل المركزية (CPU) . والنرض من هذه الوحدات الثلاث هو :

١ - وحدة الذاكرة : هي الجزء الذي تخزن فيه كل البيانات والنتائج . وتسمى الحاسبات الحالية بحاسبات البرامج المخزنة إذ أن البرامج نفسها تخزن أيضاً في الذاكرة . وتتكون وحدة الذاكرة من خلايا كثيرة جداً ، كل منها قادر على تخزين وحدة من المعلومات (كلمة) . وتسمى أيضاً هذه الخلايا ذات الطبيعة الكهرومغناطيسية ، بمراكز التخزين (متغيرات) . وترقم الخلايا بالتسلسل للفرقة بينها ويشار إلى أرقامها أحياناً بالعنوان . أما بالنسبة للمستخدم ، فيمطى هذه العناوين في شكل رموز (أسماء) .

٢ - وحدة التحكم : تتحكم في أنشطة الأجزاء المختلفة للحاسب . فهي ترسل الأوامر وإشارات التحكم وتحدد أسبقية التعليمات المختلفة .

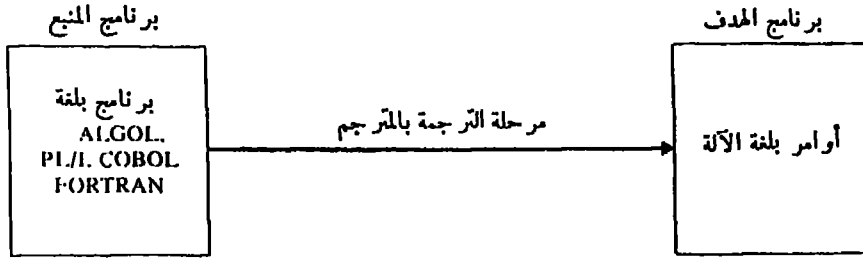
٣ - الوحدة الحسابية : تتكون من دوائر كهربائية وتقوم بالمعاملات الحسابية والمنطقية المختلفة .

(ب) الترتيب الهرمي للغات البرمجية

لكل حاسب لغة الآلة الخاصة به . ويجب أن تعطى التعليمات للحاسب بهذه اللغة حيث أنها اللغة الوحيدة التي يفهمها الحاسب . إلا أن هذه اللغة تختلف من حاسب لآخر ، وتعتمد على المكونات المادية للحاسب . وتعتمد البرامج المكتوبة بلغة الآلة على الآلة نفسها ولا تصلح إلا لما فقط .

تمثل تعليمات لغة الآلة عادة بالأعداد الثنائية ، أي سلسلة مكونة من أحساد وأصفار (1's and 0's) ، والبرمجة بلغة الآلة معقدة وتقصيلية . وفي درجة أعلى من سلم الترتيب الهرمي للغات البرمجية توجد لغة التجميع . وفيها تستبدل سلاسل الأحاد والأصفار بالرموز بحيث تعطى الأوامر بأكواد رمزية تسمى فهمونيك . وعلى ذلك نحتاج إلى برنامج يجمع لترجمة لغة التجميع إلى لغة الآلة . وبما أن تركيب لغة التجميع مشابه جداً للغة الآلة . فهذا يتطلب من مخططي البرامج أن يهتموا بالتفاصيل المتعددة مثل الفهرسة وأماكن التخزين بالإضافة إلى كتابة سلسلة من الأوامر المعقدة .

والآن نستطيع ، أن نكتب برامج للحاسب بلغة مقارنة للغة الإنجليزية مثل FORTRAN و COBOL و ALGOL و PI/I الخ . وتسمى هذه اللغات المترجم (رفيعة المستوى) وتكاد تكون مستقلة عن الآلة ، ويمكن بتعديلات بسيطة استخدام البرامج المكتوبة بهذه اللغات على الآلات المختلفة طالما وجد المترجم الخاص بها على هذه الآلة . والمترجم هو برنامج خاص يترجم تعليمات البرنامج المكتوب باللغات رقيقة المستوى إلى لغة الآلة . ويسمى البرنامج الأصل ببرنامج المنبع وترجمته تسمى ببرنامج الهدف . (أنظر شكل ١ - ٩) . من المهم أن نتذكر أن اللغات رقيقة المستوى تترجم أولاً إلى أوامر بلغة الماكينة (الآلة) قبل أن ينفذها الحاسب فعلاً .



شكل ١ - ٩

رغم أن اللغات رقيقة المستوى أقل كفاءة من لغات الآلة أو لغات التجميع ، إلا أنها تخلص مخطط البرامج من عبء الاحتفاظ بالتفاصيل الدقيقة ، مثل أماكن التخزين ، الخ ، بالإضافة إلى أنها أسهل في التعلم والاستخدام .

وكما ذكرنا سابقاً ، أن اللغة رقيقة المستوى التي ستعملها تسمى FORTRAN وستستخدم أيضاً المصطلح «حاسب» بعبارة الواسع أي متضمناً المترجم ، ووحدة الطباعة والوحدات الإضافية الأخرى .

(٢) الجمل المنفذة وغير المنفذة .

كما سبق وأن ذكرنا أنه سيتم أولاً تناول برنامج الفورتران بواسطة مترجم فورتران وهو يترجم البرنامج إلى أوامر بلغة الآلة . فبعض جمل الفورتران يقصد بها إمداد المترجم بمعلومات فقط ، ولا ينتج عنها أي أوامر بلغة الآلة . وتسمى هذه الجمل جملاً غير منفذة . بمعنى آخر « فالجمل القابلة للتنفيذ » هي تلك التي ينتج عنها نوع ما من الأوامر بلغة الآلة .

فجمل STOP جملة قابلة للتنفيذ ، والمقابل لها في لغة (الآلة) أن يتوقف الحاسب عن تنفيذ أي أمر من أوامر بلغة الآلة . التالي في البرنامج . ومن ناحية أخرى فجمل END جملة غير منفذة . فهي تخبر الحاسب أن هذه هي نهاية البرنامج ، وأنه لا توجد هناك جملة فورتران أخرى مطلوب ترجمتها إلى أوامر بلغة الآلة . وحيث أن برنامج الفورتران يترجم بالكامل قبل تنفيذ أي أمر بلغة الآلة فهذا يفسر السبب في أن جملة END هي دائماً آخر جملة في برنامج الفورتران .

هناك جملة فورتران أخرى كثيرة غير قابلة للتنفيذ سنذكرها في وقتها كلما تقدمنا في النص .

مسائل محلولة

١ - ١ حدد مخرج البرنامج التالي :

C FIRST SOLVED PROGRAM
 J = 1
 K = 3
 L = 2*J + K
 J = 3*J + 2*L
 K = K + 2
 L = J + K + L
 PRINT, J, K, L
 STOP
 END

تنفيذ البرنامج

السطر الأول : يتخطى الحاسب هذه الجملة حيث أنها تعليق .

السطر الثاني : يخزن الحاسب 1 في مكان التخزين المسمى J .

J ← 1

السطر الثالث : يخزن الحاسب 3 في مكان التخزين المسمى K

K ← 3

السطر الرابع : تأمر الحاسب بحساب قيمة $2 \cdot J + K$ وبعد ذلك تخزن النتيجة في مكان التخزين المسمى L . وعلى ذلك :

$$2 \cdot J + K = 2 \cdot 1 + 3 = 5$$

ثم يخزن الحاسب 5 في L

L ← 5

السطر الخامس : تأمر الحاسب بإيجاد قيمة $3 \cdot J + 2 \cdot L$ باستعمال القيم الحالية ل J و L وبعد ذلك يستبدل القيمة الحالية ل J بالنتيجة وعلى ذلك .

$$3 \cdot J + 2 \cdot L = 3 \cdot 1 + 2 \cdot 5 = 13$$

ثم يسمح الحاسب للقيمة الحالية ل J ويستبدلها بـ 13 ونبين هذا بالآتي :

J ← 13

السطر السادس : تأمر الحاسب بزيادة القيمة الحالية ل K بمقدار 2 .

K ← 5

السطر السابع : يسمح الحاسب للقيم الحالية ل J و K و L .

$$J + K + L = 13 + 5 + 5 = 23$$

وبعد ذلك ، يسمح الحاسب للقيمة الحالية ل L ويستبدلها بـ 23

L ← 23

السطر الثامن : يطبع الحاسب القيم الحالية لـ J و K و L ، أى :

13,	5,	23
-----	----	----

السطر التاسع : يأمر الحاسب بالتوقف STOP

الخروج

يتكون الخرج من الأعداد 13 و 5 و 23 التي تم طباعتها على سطر واحد حيث أن جملة PRINT استعملت مرة واحدة فقط .

١ - ٢ حدد خرج البرنامج التالي .

```
C
SUM OF SQUARES
K = 1
JSUM = 0
31 JSUM = JSUM + K**2
K = K + 3
IF(K.LT.10) GO TO 31
PRINT, JSUM
STOP
END
```

حدد الخرج إذا ما استبدل السطر السادس من البرنامج بما يلي :

```
IF(K.LT.20) GO TO 31
```

تنفيذ البرنامج

الخطوة :

١ - السطر الأول : يتخطى الحاسب هذه الجملة حيث أنها تعليق .

٢ - السطر الثاني : $K \leftarrow 1$

٣ - السطر الثالث : $JSUM \leftarrow 0$

٤ - السطر الرابع : يحسب الحاسب قيمة $JSUM + K**2$ (حيث $K**2$ في الفورتراي تداوى K^2) باستعمال القيم الحالية لـ JSUM و K :

$$JSUM + K**2 = 0 + 1^2 = 1$$

بعد ذلك يسمح الحاسب الصفر الموجود في SUM ويستبدله بالقيمة 1 .

$$JSUM \leftarrow 1$$

٥ - السطر الخامس : يزيد الحاسب قيمة K بمقدار 3 .

$$K \leftarrow 4$$

٦ - السطر السادس : يسأل السؤال :

$$\text{هل } K < 10 \text{ ؟}$$

وحيث أن $K = 4$ ، فالإجابة نعم ، أى أن ، الجملة بين القوسين صحيحة . ومن ثم يذهب الحاسب إلى الجملة التى تحمل رقم 31 ، أى السطر الرابع فى هذا البرنامج .

٧ - السطر الرابع : يأمر الحاسب بزيادة قيمة JSUM بمقدار K^2 . وحيث أن $K=4$ يجمع الحاسب 16 على القيمة الحالية لـ JSUM

$$JSUM \leftarrow JSUM + 16$$

٨ - السطر الخامس : $K \leftarrow K + 1$

٩ - السطر السادس : يذهب الحاسب إلى الجملة التى تحمل رقم 31 ، وهى السطر الرابع حيث أن " $K < 10$ " صحيحة .

١٠ - السطر الرابع : يجمع الحاسب 49 على القيمة الحالية لـ JSUM حيث أن $K = 7$.

$$JSUM \leftarrow JSUM + 49$$

١١ - السطر الخامس : $K \leftarrow K + 1$

١٢ - السطر السادس : بما أن $K = 10$ فالجملة

$$K < 10$$

غير صحيحة ومن ثم يذهب الحاسب إلى الجملة التالية فى البرنامج .

١٣ - السطر السابع : يطبع الحاسب القيمة الحالية لـ JSUM وهى

$$66$$

١٤ - السطر الثامن : يأمر الحاسب بالتوقف بالترتيب STOP .

النتيجة

الخرج هو العدد الصحيح 66 فقط وهى القيمة الأخيرة لـ JSUM لاحظ أن هذه القيمة النهائية لـ JSUM هى مجموع :

$$JSUM = 1^2 + 4^2 + 7^2$$

أى مجموع مربعات القيم الصحيحة فى السلسلة التى تبدأ بـ 1 وتزداد بخطوات مقدارها 3 وتنتهى قبل أن تصل إلى القيمة 10.

فى حالة ما إذا استبدل السطر السادس بما يلى :

$$IF(K.LT.20) GO TO 31$$

سيستمر المجموع إلى ما قبل 20 أى أن القيمة النهائية لـ JSUM ستكون

$$JSUM = 1^2 + 4^2 + 7^2 + 10^2 + 13^2 + 16^2 + 19^2 = 952$$

وفى هذه الحالة سيكون الخرج هو العدد الصحيح 952 .

١ - ٣ إشرح الفرض من كل برنامج من البرامج التالية :

```

READ, R
AREA = 3.1416*R*R
PRINT, R
PRINT, AREA
STOP
END

```

(أ)

```

READ, A, B, C, D
PROD = A*B*C*D
PRINT, A, B, C, D
PRINT, PROD
STOP
END

```

(ب)

(أ) يحسب هذا البرنامج مساحة الدائرة ذات نصف القطر R المعلوم .

(ب) يحسب هذا البرنامج حاصل الضرب (PROD) لأربعة أعداد مطاة هي : A و B و C و D

مسائل تكميلية

١ - ٤ أوجد الخرج للبرنامج التالي :

C FIRST SUPPLEMENTARY PROGRAM

```

I = 3
J = 2
K = 2*I + J
I = I + 3
J = I + 3*J - K
K = 2*J + 3*K
PRINT, I, J, K
STOP
END

```

١ - ٥ حدد الخرج للبرنامج التالي :

```

K = 3
JSUM = 0
51 JSUM = JSUM + K
K = K + 4
IF (K .GT. 25) GO TO 51
PRINT, JSUM
STOP
END

```

١ - ٦ أوجد الخرج للبرنامج التالي :

C THIRD SUPPLEMENTARY PROGRAM

```

I = 1
J = 2
71 K = 2*I + 3*J
I = J + K
J = I + 2*J
IF (I .GT. 100) GO TO 71
PRINT, I, J, K
STOP
END

```

١-٧ حدد المخرج للبرنامج التالي :

```

C   FOURTH SUPPLEMENTARY PROGRAM
    I = 1
    PRINT, I
    J = 2
    PRINT, J
41  K = I + J
    PRINT, K
    I = J
    J = K
    IF(K.LT.99) GO TO 41
    STOP
    END

```

١-٨ أوجد المخرج للبرنامج التالي :

```

C   FIFTH SUPPLEMENTARY PROGRAM
    J = 1
    K = 3
11  PRINT, J
12  J = J + 1
    IF(J.LT.K) GO TO 11
    K = K + 3
    IF(J.LT.20) GO TO 12
    STOP
    END

```

اجابات للمسائل التكميلية المختارة

- ١-٤ تطبع الأعداد الصحيحة 6 و 4 و 32 على سطر واحد .
- ١-٥ يطبع العدد الصحيح 78 .
- ١-٦ تطبع الأعداد الصحيحة 568 و 776 و 464 على سطر واحد .
- ١-٧ تطبع الأعداد الصحيحة : 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 على أسطر مختلفة .
- ١-٨ يطبع الحاسب الأرقام من 1 إلى 20 تاركاً الرقم 3 ومضاعفاته على أسطر مختلفة ، أى: 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20

الفصل الثاني

جمل رياضية

٢ - ١ مقدمة

لقد وضعنا في الفصل الأول كيف يقرأ الحاسب بعض برامج الفورتان البسيطة وكيفية تنفيذها ، والآآن سندرس لغة الفورتان بصورة منهجية حتى نستطيع أن نكتب برامج لحل المسائل المختلفة .

فيما يلي قائمة بالحروف المستخدمة في لغة الفورتان :

أرقام :	0	1	2	3	4	5	6	7	8	9			
حروف هجائية :	A	B	C	D	E	F	G	H	I	J	K	L	M
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
رموز :	+	-	*	/	.	,	'	=	\$	()		

هناك أيضاً المسافة الخالية والتي تعرف بحرف . سنشير إلى المسافة الخالية أحياناً بالحرف "b" ويكتب b كدليل . تسمى الأرقام مع الحروف الأبجدية ، بالحروف الأبجدية الرقية . ولا توجد حروف صغيرة .

وأحياناً يتم التفرقة بين الرقم 0 (صفر) وبين الحرف O بكتابة شرطة على الحرف O بالصورة التالية Ø .

٢ - ٢ ثوابت عددية (الأعداد)

هناك نوعان متميزان من الأعداد في الفورتان ، الأعداد الصحيحة والأعداد الحقيقية . وبصورة تقريبية ، فالاعداد الصحيحة هي تلك التي نستعملها في العد ، والاعداد الحقيقية هي تلك التي نستعملها في القياس . ويختلف الحساب والتشيل (داخليا وخارجيا) لذين النوعين من الأعداد في الفورتان . وسنناقش التشيل الخارجي الخاص بكل نوع أي الاشكال التي نكتبها ويطبعها الحاسب . بينما ستم مناقشة التشيل الداخلي أي الاشكال التي يستخدمها الحاسب لتخزين البيانات بداخل الذاكرة في الملحق (أ) .

يسمى الرقم الصحيح أيضاً الثابت ذو النقطة الثابتة وهو أي عدد صحيح له أو ليس له إشارة جبرية وبدون علامة عشرية أو أي علامات وقف أخرى . وسنعتبر الأرقام بدون الإشارة الجبرية أرقاماً موجبة . وأقصى طول لأي رقم صحيح لا يتعدى عادة 9 خانات ، غير إنها تختلف من حاسب إلى آخر . وننصح مخططي البرامج بفرص إمكاناتهم الحسابية المحلية لمعرفة تلك الصفات التي تعتمد على الآلة .

مثال ٢ - ١

الاعداد التسالية مقبولة كثوابت صحيحة .

5286 - 12547 0 -8 15

بينما الأعداد التسالية غير مقبولة كثوابت صحيحة :

(يحتوي على علامة عشرية)	21.0
(يحتوي على علامة عشرية)	— 248.
(يحتوي على فاصلة)	12,357
(كبير جداً)	125000000000

يمكن أن نكتب الثوابت الحقيقية في الفورتان ، والتي تسمى أيضاً بثوابت التقلبة الطليقة في شكلين مختلفين : الشكل العشري والشكل الأسى . في كلا الشكلين توجد سلسلة محددة من الخانات مع العلامة العشرية . وفي الواقع ، فإن العلامة العشرية هي التي تميز الثابت الحقيقي عن الثابت الصحيح .

الشكل العشري : يكتب الثابت الحقيقي في الشكل العشري كسلسلة محددة من الخانات سواء بإشارة جبرية أو بدونها ، مع علامة عشرية . ولا يمكن أن توجد أى علامة وقف أخرى . من الممكن كتابة الرقم الحقيقي بأى عدد من الخانات المنوية ، إلا أن ، عدد الخانات المنوية التي يحتفظ بها الحاسب تتوقف على طول كلمة الحاسب . وبصورة عامة ، لا يحتفظ الحاسب بأكثر من سبعة أو ثمانية أرقام منوية . مرة أخرى ، يجب على القراء مراجعة إمكاناتهم الحسابية لهذه الخصائص التابعة للآلة .

مثال ٢ - ٢

الأعداد التسالية مقبولة كتوابت حقيقية :

$$28.3 \quad -236.0 \quad 24. \quad +.0187 \quad -234.182$$

بينما الأعداد التسالية غير مقبولة كتوابت حقيقية :

$$. (\text{يحتوي على فاصلة}) \quad 4,356.2$$

$$. (\text{لا توجد علامة عشرية}) \quad -36$$

الشكل الأسى : تتكون الثوابت الحقيقية عندما تكتب في الشكل الأسى (E-Form) من جزئين . الجزء الأول هو ثابت حقيقى في الشكل العشري (أى سلسلة محددة من الخانات سواء بإشارة جبرية أو بدونها مع علامة عشرية) . ويبدأ الجزء الثانى بالحرف E يتبعه ثابت صحيح سواء بإشارة جبرية أو بدونها ومكون من خانتين على الأكثر . ويفسر الجزء الثانى على أنه الأس للأساس 10 . على سبيل المثال :

$$23.1E-4$$

هو ثابت حقيقى في الشكل الأسى (E-form) ويمثل 23.1×10^{-4} وفيما يلي أمثلة مقبولة لثوابت حقيقية في الشكل الأسى (E-form)

$$24.123E04 \quad 24.123 \times 10^4 = 241230.0$$

$$-5.36E-3 \quad -5.36 \times 10^{-3} = -0.00536$$

$$. \quad 2.6E+2 \quad 2.6 \times 10^2 = 260.0$$

لا توجد أى قيود بخصوص مكان وضع العلامة العشرية عندما نكتب أو ندخل عدداً حقيقياً في الشكل الأسى (E-form) . ومن ثم ، فيمكن كتابة أى ثابت حقيقى في الشكل الأسى (E-form) بأكثر من طريقة ، على سبيل المثال :

$$23.1E-4 \quad 2.31E-3 \quad 0.0000231E2$$

كلها . نفس الرقم ويقبلها الحاسب .

من الممكن دائماً أن يمثل الرقم الحقيقي في الشكل الأسى المعيارى أى أن الرقم السابق للحرف E (المسمى بالجزء الحقيقي) يقع بين :

$$0.1 \text{ و } 1.0 \quad (\text{أو } -0.1 \text{ و } -1.0)$$

وفى واقع الأمر سوف يطبع الحاسب دائماً (أى الخرج) الثابت الحقيقي في الشكل الأسى (E-form) بهذه الطريقة .

وكما ذكرنا سابقاً . فإن عدد الخانات المنوية التى يحتفظ بها الحاسب تتوقف على الآلة . بالإضافة إلى ذلك ، فإن الحد الأقصى المقبول للأس (عادة ما يحدد بمخاتنين على الأكثر) و: يتوقف الآلة أيضاً .

ملاحظة : يختلف تصور الثابت الحقيقي في الفورتران عنه في الرياضيات . ففى الواقع فإن كل الأرقام الحقيقية في الرياضيات تقرب بأعداد منطقية في الحاسبات .

رغم أن 24.0 و 2.4E1 مختلفان في المظهر (الخارجى) إلا أنهما متطابقان في التمثيل الداخلى حيث أن كليهما أعداد حشرية (انظر ملحق أ) . ومن ناحية أخرى 24 و 24.0 لها تمثيل داخلى مختلف ويميلان بصورة مختلفة في الفورتران فأحدهما يدل على ثابت صحيح والآخر يدل على ثابت حقيقى (كلاهما يمثل نفس العدد رياضياً) .

لم نناقش حتى الآن ، كيف نأمر الحاسب أن يقرأ أو أن يطبع عدداً صحيحاً و/أو عدداً حقيقياً . ستناقش هذه الأمور في الفصل الثالث.

٢ - ٣ اسماء المتغيرات (اسماء اماكن التخزين)

يستخدم دائماً المصطلح « متغير » في علم الحاسب ليعنى عنصر ذاكرة . والخسائص الأساسية التى يجب أن نتذكرها عن عناصر الذاكرة هى : القراءة الهدامة إلى الذاكرة والقراءة غير الهدامة من الذاكرة . أى أنه إذا خزننا معلومات في خلية ذاكرة فسوف تتبدد المحتويات الأصلية (السابقة) في خلية الذاكرة ولكن إذا نقلت (نسخت) معلومات من الخلية فستبقى المحتويات (الأصلية) في الخلية كما هى بدون تغيير .

إذا استعمل مكان ذاكرة (متغير) لتخزين ثوابت صحيحة ، سمي متغيراً صحيحاً أما إذا استعمل لتخزين ثوابت حقيقية سمي متغيراً حقيقياً . تعطى الاسماء لخلايا الذاكرة ، لسهولة الرجوع إليها . وعموماً يرمز لأسم المتغير بواسطة عدد من الحروف الأندية تتر اوح ما بين حرف إلى ستة حروف ، ويجب أن يكون الحرف الأول حرفاً أبجدياً ، وعلى ذلك فالأسماء: IAM, A2X, DAD, DEAN . المتغيرات مقبولة أما الأسماء التالية فهى أسماء غير مقبولة .

2XY لأنها لا تبدأ بحرف أبجدي .

A*B6 لأن * ليست حرفاً أبجدياً رقياً .

INTEREST تحتوى على أكثر من ستة حروف .

(تقبل بعض الحاسبات أسماء للمتغيرات أكثر من ستة حروف ، ولكن بالنسبة لمعظم الحاسبات فالحد الأقصى هو ستة حروف .)

يستعمل الفورتران أيضاً القاعدة الضمنية الآتية لأسماء المتغيرات : إذا كان أول حرف هو أحد الحروف الستة التالية I, J, K, L, M, N فهو اسم لمتغير صحيح ، أما إذا كان الحرف الأول أى حرف من الحروف الأخرى ، أى أحد الحروف التالية A, ..., H, O, P, ..., Z فهو اسم لمتغير حقيقى (يمكن إبطال هذه القاعدة بجمللة اعلانية النوع التى ستناقش في القسم القادم) .

مثال ٢ - ٢

المتغيرات التالية مقبولة كمتغيرات صحيحة :

MONEY NEXT J4X2 IDIOT

بينما المتغيرات التالية مقبولة كتغيرات حقيقية :

ANSWER X21 BUM ENEMY

ملاحظة : كلمات معينة مثل Pi.AD و WRITE و PRINT و PAUSE و STOP و END الخ ، هي كلمات دالة في لغة الفورتران ولذلك لا تقبل كأسماء . - ات في معظم الحاسبات .

٢ - ٤ جمل النوع - صحيح وحقيقي

في بعض الأحيان يكون غير مرغوب أن نتبع القاعدة السابقة لكتابة المتغيرات . فعل سبيل المثال قد ترغب في استخدام (الرمز التيموني - مساعد للذاكرة) INT ليرمز إلى متغير حقيقي لسعر الفائدة . فلإبطال القاعدة الفسنية يمدنا الفورتران بعدة جمل صريحة للإعلان عن النوع . فجملة الفورتران المبتدئة بكلمة REAL متبوعة بقائمة من المتغيرات والتي تفصل بين كل منها وصلة (.) تملن أن هذه المتغيرات REAL (أي متغيرات حقيقية) . وعلى ذلك فإن

REAL INT

تملن الحاسب بأن INT ترمز دائماً إلى متغير حقيق طوال البرنامج ، وبطريقة مشابهة فإن جملة الفورتران المبتدئة بكلمة INTEGER تتبعها قائمة من أسماء المتغيرات تفصل بين كل منها فصلة (و) ، تملن أن هذه المتغيرات INTEGER (أي متغيرات صحيحة) .

تمد جمل النوع المترجم بمعلومات ، ومن ثم فهي جمل غير منفذة . ويجب أن تظهر في بداية البرنامج . وسبب تقديم جمل نوع أخرى غير INTEGER و REAL في فصول قادمة .

مثال ٢ - ٤

افترض أن برنامجاً يبدأ بالجمل التالية :

INTEGER A, B, SUM, TIME
REAL NUMBER, MIN, MAX

وعلى ذلك ، فإن البرنامج سيشير A و B و SUM و TIME إلى متغيرات صحيحة أما NUMBER و MIN و MAX مستشير إلى متغيرات حقيقية .

يشير بعض منظمي البرامج بأن ذكر كل المتغيرات في جمل نوع بنفس النظر عن نوعية الحروف الأول من هذه المتغيرات . تمرين يستحق التأمل فعل سبيل المثال ، يمكننا كتابة :

REAL NUMBER, MIN, MAX, RATE

رغم أن RATE متغير حقيق إن لم يكتب صراحة في جملة نوع .

٢ - ٥ عمليات حسابية - حسابات صحيحة وحقيقية

يبين الجدول ٢ - ١ الرموز الخمس للعمليات الحسابية الأساسية في الفورتران ونحن نؤكد أن التعبيرات الجبرية :

$$a \cdot b \quad \frac{a}{b} \quad a^b$$

تكتب : $A \cdot B, A/B, A \cdot \cdot B$ على الترتيب .

جدول ٢ - ١

الرمز	العملية
+	جمع
-	طرح
*	ضرب
/	قسمة
**	الرفع الأس

حيث أن هناك نوعين من الأرقام ، صحيحة وحقيقية فهناك كذلك نوعان من الحسابات حسابات حقيقية وحسابات صحيحة ويتم إنجاز هذين النوعين من الحسابات بطرق مختلفة ولذا فن المهم جداً أن يكون الفرق بينهما مفهوماً فهما واضحاً .

(أ) حسابات صحيحة

إذا كانت المعاملات أعداداً صحيحة ، تنفذ الحسابات الصحيحة لتنتج عدداً صحيحاً . على سبيل المثال ، تستخدم الحسابات الصحيحة لإيجاد قيمة : $5 + 3, 5 * 3, 5 - 3$

فيكون الناتج 8 و 15 و 2 على الترتيب . وفي القسمة الصحيحة في الفورتران عند إيجاد قيمة I/J حيث كل من I و J عدد صحيح يكون الناتج أيضاً صحيحاً وهو الجزء الصحيح من خارج القسمة . وعلى ذلك يتر الجزء الكسرى من خارج القسمة في القسمة الصحيحة . على سبيل المثال :

$$5/4 \text{ تعطى } 1 \text{ و } 4/5 \text{ تعطى } 0 \text{ و } 7/2 \text{ تعطى } 3 \dots$$

وهكذا فالقسمة الصحيحة في الفورتران تختلف تماماً عن القسمة العادية ، ومع ذلك يمكن استخدامها في صالحنا . كما هو موضح في المثال ٢ - ٥ .

(ب) حسابات حقيقية

إذا كانت المعاملات حقيقية ، تنفذ الحسابات الحقيقية لتنتج قيماً حقيقية . وعلى ذلك فالحسابات الحقيقية تستخدم لإيجاد :

$$5. + 3, 5. * 3, 5. - 3.$$

لتعطينا القيم الحقيقية 8. ، 15. ، 2. على الترتيب . والقسمة بالحسابات الحقيقية في الفورتران مشابهة للقسمة العادية أي أن :

$$5./4 \text{ تعطى } 1.25 \text{ و } 4./5 \text{ تعطى } 0.8, 7./2 \text{ تعطى } 3.5 \dots$$

أهم جزء في المناقشة المذكورة اعلاه هو أن نوع المعامل يحدد نوع الحساب (أو نمط التنفيذ) ويكون الناتج محصلة من نفس النوع . ونلاحظ أن الرفع للأس يعامل بطريقة مختلفة . فالقيم الحقيقية الموجبة فقط هي القيم التي يمكن أن ترفع إلى قوى حقيقية بينما أي قيمة سواء صحيحة أو حقيقية يمكن أن ترفع إلى قوى صحيحة . وسنناقش ذلك في القسم ٢ - ٨ .

مثال ٢ - ٥

مع فرض أن الاستخدام التقليدي للأقواس في التعميرات الحسابية مقبول أيضاً في الفورتران . ناقش الفروق بين :

$$(أ) (N+2)/2 \quad \text{و} \quad (ب) (N/2)+2$$

بين أيضاً كذا ، يمكن أن تستخدم (ب) لتحديد ما إذا كانت N زوجية أو فردية .

التعبيرات الجبرية التي تمثلها هي :

$$\frac{n \cdot 2}{2} \quad \text{و} \quad \frac{n \cdot 2}{2}$$

على الترتيب : وهما متطابقتان رياضياً . ولكنهما ، بالحساب الصحيح في الفورتران غير متكافئتين . وبالتحديد ، لدينا الآتي :

(أ) هنا نحسب قيمة $N+2$ أولاً لتعطي عدداً زوجياً صحيحاً ، ومن ثم $(N+2)/2$ دائماً ستأخذ القيمة N .

(ب) هنا نحسب قيمة $N/2$ أولاً لتعطي الجزء الصحيح من خارج القسمة وعلى ذلك إذا كانت N زوجية ، فتعطي $(N/2)+2$ الناتج

N ولكن إذا كانت N فردية فتعطي $(N/2)+2$ الناتج $N \pm 1$. على سبيل المثال إذا كانت $N = 8$ فإن :

$$(N/2)+2 = 4+2 = 6$$

وإذا كانت $N = 11$ فإن :

$$(N/2)+2 = 5+2 = 7$$

وإذا كانت $N = 9$ فإن :

$$(N/2)+2 = 4+2 = 6$$

٢ - ٦ التعبيرات الحسابية

يجب أن نفهم كيف يقوم الحاسب بإيجاد قيمة التعبيرات الحسابية حتى نستطيع كتابتها بطريقة صحيحة .

يوجد الحاسب قيمة التعبيرات الحالية من الأقواس (أى ، التعبير أو التعبير الفرعى بدون أقواس) ، باستعمال جدول الأسبقية

التقليدى التالي :

أسبقية أول : الأس (**)

أسبقية ثانية : الضرب (*) والقسمة (/)

أسبقية ثالثة : جمع (+) وطرح (-)

(ستعامل الزائد الأحادى والناتقص على نفس المستوى مثل الجمع الثنائي والطرح أى $2.2**2 - 2.2**2$ تعنى $(2.2**2) -$ والتي تنتج 4.84) .

حيث أن هناك ثلاثة مستويات للأسبقية ، فتصور أن الحاسب يسمح (أو يمر على) التعبيرات الرياضية الحالية من الأقواس من اليسار إلى اليمين ثلاث مرات . المرة الأولى للبحث عن الأس ؛ والمرة الثانية للبحث عن الضرب والقسمة ، وأخيراً ، للبحث عن الجمع والطرح . حيث أن الحاسب يمر من اليسار إلى اليمين ، فإذا كانت عمليتان لهما نفس الأسبقية ، فتتخذ أولاً تلك التي على أقصى اليسار .

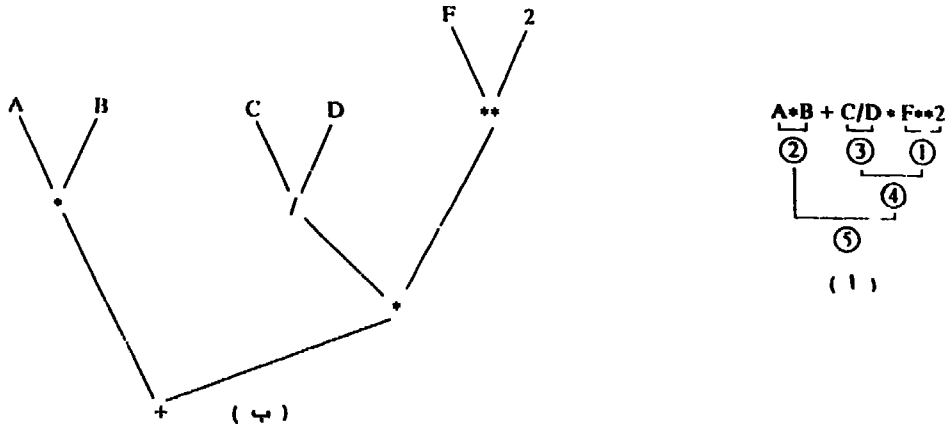
مثال ٢ - ٦

إدرس التعبير

$$A*B + C/D**2$$

يوضح شكل ١-٢ ترتيب التنفيذ . ويمكن لهذا الترتيب أن يكتب بصورة مكافئة كشجرة مقلوبة والمعاملات الحسابية كمقد (nodes) كما هو موضح في شكل ١ - ٢ (ب) . لاحظ أن التعبير يمثل ما يلي :

$$a \cdot b + \frac{c}{d} \cdot f^2$$

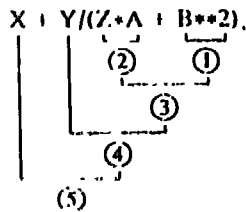


شكل ١ - ٢

يسمح الـ تران أيضاً استخدام الأقواس بالطريقة التقليدية . بمعنى أن الأقواس لها أسبقية أعلى من كل العمليات الحسابية ، وتعامل الأقواس الداخلية أولاً .

مثال ٢ - ٧

(أ) يمكن إيجاد قيمة التعبير $(X + Y/(Z * A + B**2))$ كالتالي :



وهكذا فالتعبير الجبري هو :

$$x + \frac{y}{z \cdot a + b^2}$$

(ب) المعادلة الجبرية :

$$x = [(a + b)^2 + (3c)^3]^{ab}$$

يمكن أن تكتب بالفورتران كالتالي :

$$X = ((A + B)**2 + (3.0*C)**3)**(A/B)$$

ونصح بشدة باستخدام أقواس إضافية كلما كان هذا الاستعمال مؤكدا للصواب . علاوة على ذلك ، إذا استوى أى تعبير رياضي على حدود طويلة ومعقدة ، فتجنب دائماً تقسيمها إلى عدة جمل أصغر وأقل عرضة للخطأ . على سبيل المثال التعبير الجبري :

$$y = \left[\frac{ab}{c+d} - \frac{g}{5(h+x)} \right]^{1/r}$$

يجب أن تمثل كما يلي :

$$T1 = A*B/(C + D)$$

$$T2 = G/(5*(H + X))$$

$$Y = (T1 - T2)**(1./R)$$

ملاحظة : لا يسمح الفورتران بظهور عمليتين حاسبتين متجاورتين . وعلى هذا B - A* يجب أن تكتب (-B) * A*

مثال ٢ - ٨

فيما يلي بعض تعبيرات رياضية وتميرات الفورتران المقابلة لكل :

فورتران	رياضيا	
$a + \frac{b}{c} + d$	$A + B/C + D$	(١)
$\frac{a+b}{c+d}$	$(A + B)/(C + D)$	(٢)
$a^3 - b^3$	$A**3 - B**3$	(٣)
$\frac{ab}{c-d^2}$	$A*B/(C - D**2)$	(٤)
$\frac{ab}{c} \cdot d^2$	$A*B/C - D**2$	(٥)
$\frac{a}{bc} - d^2$	$A/(B*C) - D**2$	(٦)
$1 + \frac{a}{b + \frac{1}{c}}$	$1. + A/(B + 1./C)$	(٧)

لاحظ ضرورة وجود الأقواس في (و) ولكنها غير ضرورية في (٥) .

٢ - ٧ العمليات الحسابية ذات النمط المختلط

وإذا ناقشنا في القسم ٢ - ٥ ، حيناً يحسب الحاسب قيمة تعبير رياضي فإن نوع حدود التعبير تحدد نوع الحساب المستخدم (نمط العملية) . وعلى ذلك ، يجب أن تكون كل المتغيرات والثوابت في أي تعبير رياضي بالفورتران من نفس النوع ، أي يجب أن تكون جميعها إما حقيقية أو صحيحة . وبالتالي إذا كانت :

$$A + 3$$

هو تعبير أحسد حدوده A متغير حقيقي والأخر صحيح ، فيمكن أن نتساءل :

(١) عما إذا كان مسموح بمثل هذا النمط المختلط في التعبير .

(٢) وحتى إذا كان مسموحاً به فما هي نوعية الحسابات التي سوف تنفذ .

لا تسمح بعض مترجمات (Compilers) الفورتران بالتعبيرات الرياضية ذات النمط المختلط . في مثل هذه الحالة ، يجب أن تحتاط لتجنب الخلط بين نوعين في تعبير واحد . وبذلك فالتعبيرات

$$\begin{array}{lll} A + 3 * B & \text{يجب أن تكتب} & A + 3 * B \\ I + 4 * J & \text{يجب أن تكتب} & I + 4.0 * J \\ B ** 2 - 4. * A * C & \text{يجب أن تكتب} & B ** 2 - 4 * A * C \end{array}$$

لا يعتبر رفع رقم حقيقي إلى قوة صحيحة نمط عمليات مختلط (أنظر المناقشة في نهاية قسم ٢ - ٨) ، وعلى هذا $B ** 2$ يسمح بها ولا يجب أن تغير إلى $B ** 2.0$ (لمعرفة الفرق بين $B ** 2$ و $B ** 2.0$ أنظر قسم ٢ - ٨ وأيضاً المسألة ٢ - ١٧) .

أغلب انسياس الكبرى لديها مترجمات تقبل تعبيرات النمط المختلط . في مثل هذه الحالة ، إذا كانت حدود إحدى العمليات من نوعين مختلفين أي إحداها صحيح والآخر حقيقي ، فيحول الصحيح أولاً إلى حقيقي ثم تجرى الحسابات الحقيقية لإعطاء قيمة حقيقية . وقد ظهر هذه وسيلة مريحة حيث يمكن ظهور كلا من المتغيرات والثوابت الصحيحة والمتغيرات والثوابت الحقيقية بعد ذلك في تعبير رياضي واحد ، ولكن يجب على مخططي البرامج أن يتحسروا من عواقب ظهور كلا النوعين من المتغيرات والثوابت في تعبير واحد .

مثال ٢ - ٩

(أ) أدرس التعبير الرياضي ، بفرض أن حساب النمط المختلط مسموح به

$$(11/2) + 4.3$$

وحيث أن 11. و 2 نوعان مختلفان فيحول العدد الصحيح 2 إلى 2. ويجرى الحسابات الحقيقية وينتج عنها القيمة الحقيقية 5.5 وجميع الأرقام الحقيقية 5.5 و 4.3 فإننا نحصل على النتيجة 9.8 .

(ب) أدرس التعبير الرياضي :

$$(11/2) + 4.3$$

نحسب قيمة الحد $11/2$ أولاً . وحيث أن 11 و 2 أعداد صحيحة ، من ثم نحسب قيمة $11/2$ بواسطة الحساب الصحيح فتكون النتيجة 5 (وليس 5.5) . وحيث أن حدود الجمع من أنواع مختلفة ، فالعدد الصحيح 5 إلى عدد حقيقي ويضاف إلى الثابت الحقيقى 4.3 باستعمال الحساب الحقيقى مما يعطى العدد الحقيقى 9.3 .

يجب أن يدرك مخططو البرامج أن وسيلة الراحة هذه لاتتأق « بدون مقابل » . كلا ، فن كل مرة نحول قيمة صحيحة إلى حقيقية سنهلك فيها وقت الآلة . وبذلك فالسلاح بالتعبيرات الرياضية ذات النمط المختلط سيزيد من وقت التنفيذ ، وسيمرض البرنامج إلى عنق الاختلال من كفاءة التنفيذ . بالإضافة إلى ذلك ، فالبرامج التي تكتب بالنمط المختلط أقل تداولاً . أى قد لا تستطيع تشغيلها على حاسبات مختلفة . وأخيراً فاستخدام النمط المختلط بدون تفكير قد ينتج قبا غير مقصوده كما هو موضح ضمناً في مثال ٢ - ٩ .

٢ - ٨ الدوال الرياضية البينية (المبنية ذاتياً)

للغة الفورتران مكتبة للدوال الرياضية التي يمكن أن يستخدمها مخطط البرامج . بعض هذه الدوال الأكثر شيوعاً معروسة في الجدول ٢ - ٢ . وهناك قائمة أخرى أكثر شمولاً مطبوعة في المجلد ١ .

جدول ٢ - ٢

الدالة	الرمز الرياضي	في الفورتران
الجذر التربيعي للمتغير X	\sqrt{x}	SQRT(X)
القيمة المطلقة للمتغير X	$ x $	ABS(X)
الرفع إلى اس X	e^x	EXP(X)
جيب الزاوية X	$\sin x$	SIN(X)
جيب تمام الزاوية X	$\cos x$	COS(X)
لوغار بسم X	$\log_{10} x$	ALOG10(X)
اللوغاريم الطبيعي ل X	$\log_e x$	ALOG(X)
تحويل العدد الصحيح I إلى عدد حقيقي		FLOAT(I)
بتر العدد الحقيقي X إلى عدد صحيح .		IFIX(X)

رغم أننا سنناقش دوال البرامج الفرعية بالتفصيل في الفصل السابع إلا أننا سوف نشرح هنا بعض الرموز والمصطلحات . فبعض التعبيرات

SQRT(X)

القيمة التي تخصصها دالة الجذر التربيعي للمتغير X . وتسمى X خلاصة الدالة ، ونطلق على SQRT اسم الدالة . لاحظ أن اسم كل دالة يتبعه قوسين يحيطون بخلاصة الدالة .

يستخدم جدول ٢ - ٢ كلا من X و I كخلاصات للدوال . تشير X إلى أن الخلاصة يجب أن تكون حقيقية ، كما تشير I إلى أن الخلاصة يجب أن تكون من النوع الصحيح . وفي الجانب الآخر يتحدد نوع قيمة الدالة بواسطة اسم الدالة تبعاً للقوانين العادية لأسماء لتسمية المتغيرات وعلى سبيل المثال :

- ALGO () تدل على قيمة حقيقية مناظرة للخلاصة الحقيقية X .
- IFIX(X) تدل على قيمة صحيحة مناظرة للخلاصة الحقيقية X .
- FLOAT(I) تدل على قيمة حقيقية مناظرة للخلاصة الصحيحة I .

وهكذا :

فالدوال FLOAT و IFIX ليست دوال رياضية عادية . تحول دالة FLOAT القيمة الصحيحة إلى القيمة الحقيقية المكافئة لها . فمثل سبيل المثال :

تعطى FLOAT(4) القيمة 4.0 وتعطى FLOAT(-25) القيمة -25.0 .
وآستخدم دالة FLOAT عادة لتجنب استخدام حساب النمط المختلط على سبيل المثال : إذا كانت SUM تشير إلى مجموع عدد N من الأرقام ، حينئذ يمكن أن نكتب :

$$AVE = SUM/FLOAT(N)$$

وذلك للحصول على المتوسط AVE لعدد N من الأرقام . تحول دالة IFIX قيمة حقيقية إلى قيمة صحيحة بواسطة البتر ، فمثل سبيل المثال ،

$$\text{تعطى IFIX(2.9) القيمة 2 وتعطى IFIX(-3.7) القيمة -3}$$

وأنتا لتؤكد أن دالة IFIX لا تقرب القيم الحقيقية إلى أقرب قيمة صحيحة .

يجب أن تأخذ في الاعتبار الملاحظات التالية :

١ - لا تستلزم أن تكون خلاصات الدالة متغيرات مفردة . ولكن يمكن أن تكون تعبيرات حسابية من النوع المطلوب . لثلا ترجمة :

$$\sqrt{b^2 - 4ac} \quad \text{بالفورتران ستكون} \quad \text{SQRT(B**2 - 4.*A*C)}$$

٢ - يمكن أن تظهر الدالة كجزء من أى تعبير حسابي طالما نعطي الصلاصة المناسبة . فثلا ترجمة :

$$ABS(A - B) + C/SIN(D) \quad \text{بالفورتان ستكون} \quad |a - b| + \frac{c}{\sin d}$$

٣ - يمكن أن تستدعى خلاصة الدالة دالة أخرى ، أى يمكن أن يكون لدينا دوال لدوال ، فثلا ترجمة :

$$SIN(ABS((X - Y)/(X + Y))) \quad \text{بالفورتان ستكون} \quad \sin \left| \frac{x - y}{x + y} \right|$$

٤ - لإيجاد قيمة التعبيرات الحسابية الخالية من الأقواس والتي تتضمن دوالا مكتوبة سبقة (مبنية داخليا) يكون لحساب قيمة هذه الدوال أسبقية على القوانين التي تمت مناقشتها في قسم ٢ - ٦ . فثلا ، التعبير :

$$\frac{ABS(A - B) + C/SIN(D)}{\quad}$$

سيتم حساب قيمته بالترتيب الموضح . لاحظ أن خلاصة الدالة ABS تعبير ، ومن ثم فنجد حساب قيمة الدالة سوف تطبق قوانين الأسبقية مرة ثانية .

مثال ٢ - ١٠

فيما يلي تعبيرات رياضية وتعابير الفورتران المناظرة لها

فورتان	رياضيا	
$SIN(X)/(ABS(Y) + COS(Z))$	$\frac{\sin x}{ y + \cos z}$	(أ)
$EXP(X + Y)/(X + Y)$	$\frac{e^{x+y}}{x + y}$	(ب)
$EXP(ABS(X - Y)) + X$	$e^{ x-y } + x$	(ج)
$A + B/ABS(FLOAT(M - N))$	$a + \frac{b}{ m - n }$	(د)

خطأ شائع يقع فيه كثير من مخططي البرامج المبتدئين عند كتابة الثابت π . وحيث أن π ليس ثابتاً ميبثاً ، وكذلك ليس حرفاً مقبولاً في الفورتران ، فيجب استخدام قيمة مقربة لـ π . فثلا ، تعبير الفورتران :

$$SIN(2.*N*3.14 + X) \quad \text{سيكون} \quad \sin(2n\pi + x)$$

حيث تستخدم 3.14 بدلا من π . وبالمثل ، يمكن أن تخصص 3.14 للمتغير PI (أنظر قسم ٢ - ٩) وبعد ذلك نكتب :

$$SIN(2.*N*PI + X)$$

نهى هذا القسم بالنقاط البسيطة الهامة التالية ، والمتدا . ارفع إلى أس حيث أنها تتضمن الدالة اللوغاريتمية .

١ - حين ترفع قيمة لقوة حقيقية كما في $X * Y$ فإن قيمتها تحسب داخليا كما يلي :

$$e^{y \ln(x)}$$

حيث \ln هي دالة اللوغاريتم الطبيعي ، أى ($EXP(Y * A * LOG(X))$. (أنظر جدول ٢-٢) - وحيث أن خلاصات الدالة اللوغاريتمية يجب أن تكون حقيقية وموجبة ، لذا قيمة X التي يمكن رفعها إلى قوى حقيقية يجب أن تكون حقيقية موجبة . ويجب أن يحتاط مخطوط البرامج بصفة خاصة لهذا حيث أن خطأ رفع قيمة سالبة إلى قوة حقيقية يكتشف فقط أثناء وقت التنفيذ .

٢ - عندما يرفع رقم إلى قوة صحيحة كما في $X ** I$ فإن ذلك ينفذ بالضرب . ومن ثم ، فيمكن رفع أى قيمة ، صحيحة أو حقيقية ، سالبة كانت أو موجبة ، إلى قوة صحيحة (أيضاً ، أنظر المسألة ٢ - ١٧) .

٣ - رياضياً ، a^{bc} تعني $a^{(bc)}$ وهى تختلف عن $(a^b)^c$. ومع ذلك فإن $A ** B ** C$ يمكن أن تأخذ الصورة $(A ** B) ** C$ فى بعض المترجمات والصورة $A ** (B ** C)$ بواسطة مترجمات أخرى . ومن ثم فنحن نشجع مخطوطي البرامج على استخدام الأقواس عند كتابة مثل هذه التعبيرات . وعموماً فن الحكمة استخدام الأقواس ، كلما كان هناك شك .

٢ - ٩ جملة التخصيص الحسابية

لكي ينفذ الحاسب تعبيراً رياضياً ، يجب أن تعطي كل المتغيرات التي تظهر في التعبير قيماً في مكان ما بالبرنامج ولكن قبل استخدامها ، أى يجب أن تعرف مسبقاً . تعطي معظم الحاسبات رسالة خطأ إذا لم يعرف أى متغير في تعبير . كيف نعرف المتغير؟ أى كيف نخزن قيمة في الذاكرة؟ يعرف المتغير أما بقراءة قيمة من مجموعة بطاقات البيانات (أنظر الباب الثالث) أو بجملة تخصيص حسابية .

رمزياً ، نشير إلى تخزين (تخصيص) الثابت 1.5 في مكان يسمى X كما يلي :

$$X \leftarrow 1.5$$

وتسمى هذه جملة تخصيص . في الفورتران \leftarrow تمثل بعلامة = ، وعلى ذلك $X \leftarrow 1.5$ تكتب في الفورتران كما يلي :

$$X = 1.5$$

من ناحية أخرى ، نفرض أن X تم تعريفها فعلاً ، ونريد حساب قيمة X^2 ثم تخزين الناتج في المكان Y أى $Y \leftarrow X^2$ في الفورتران ، يمكن أن نكتب

$$Y = X ** 2$$

وعموماً ، تأخذ جملة التخصيص الحسابية في الفورتران الشكل التالي :

تعبير حسابي (رياضى) = اسم متغير

وأننا لنؤكد أن الجانب الأيسر من علامة = يجب أن يكون إسم لمتغير مفرد .

من سوء الحظ أن الرمز = يستخدم لجمل التخصيص في الفورتران ، حيث أن = تستخدم تقليدياً لتعني التساوى . وأننا ننصح القراء بشدة أن تتذكر دائماً أن = تعني « تخصيص » أو « إحلال » أو « نسخ » ، الخ . في مناقشتنا سنستعمل أحياناً الرمز \leftarrow للتخصيص فضلاً عن بديله في الفورتران = حين نجد أن هذا الاستخدام يسهل عملية الفهم .

ملاحظة : أنه لغاية في الأهمية أن نعرف أن فعلين منفصلين يحدثان في جملة التخصيص الحسابية :

١ - إيجاد قيمة التعبير الحسابي على الجانب الأيمن من التساوى .

٢ - تخصيص نتيجة (١) لمكان التخزين على الجانب الأيسر من علامة التساوى . من المقبول أن يكون جانبا علامة = من أنواع

مختلفة . حينها يكون جانبا علامة = من أنواع مختلفة ، فيتضمن التخصيص في الخطوة الثانية تحويل النتيجة في (١) إلى نفس نوع مكان التخزين الموضح على يسار علامة التساوى . وبالتحديد ، سنعتبر القيمة الحقيقية قبل أن تخصص إلى مكان تخزين صحيح .

يختلط الأمر على كثير من مخططي البرامج المبتدئين حول التعميرات ذات النمط المختلط وجمل التخصيص ذات الأنواع المختلفة على جانبي علامة = . ويشير النمط المختلط إلى تعبيرات حسابية حيث المعاملات من أنواع مختلفة . كما تشير جمل التخصيص إلى حركة تخزين قيمة في الذاكرة ، وهى أمر فورتران ويمكن أن يشمل حساب قيمة تعبير حسابي . والتعبير الحسابي نفسه ليس أمراً . لا تقبل بعض المترجمات التعميرات ذات النمط المختلط . ولكن تقبل كل المترجمات حمل التخصيص بأنواع مختلفة على جانبي علامة = .

مثال ٢ - ١١

(أ) أدرس الجملة :

$$INCOME = PRIN * (1. + XINT)$$

التعبير الحسابي على الجانب الأيمن من = من النوع الحقيقى . تحول نتيجة حسابات الجانب الأيمن أولاً إلى صحيح (بيتر الجزء الكسرى منها) قبل تخزينها في INCOME حيث INCOME متغير صحيح . وبالتحديد ، إذا كانت $XINT = 0.0625$ و $PRIN = 100.00$ فالنتيجة $PRIN * (1. + XINT)$ هى 106.25 ولكن قبل أن (تنسخ) في INCOME بيتر الجزء الكسرى 0.25 وتخزن القيمة الصحيحة 106 في INCOME .

(ب) أدرس الجملة :

$$ASSET = INCOME + I.OOT$$

يجب هنا: التعبير الحسابي في النمط الصحيح ويحول بعد ذلك إلى النوع الحقيقى قبل أن يخزن في المكان الحقيقى المسى ASSET .

(ج) أدرس الجملة :

$$SUM \quad SUM + X$$

حيث أن هناك حركتين منفصلتين في جملة التخصيص $SUM + X$ تنفذ أولاً . ومن ثم تجمع القيم الحالية لـ SUM و X وتخزن النتيجة حينئذ في SUM مرة أخرى . وحقيقة فإن الجملة تقول :

« القيمة الجديدة لـ SUM هى نتيجة جمع القيمة القديمة لـ SUM و X »

من الواضح ، أنه إذا ترجمت العلامة = على أنها « تساوى » فالجملة المعطاة $SUM = SUM + X$ ستكون بلا معنى على الإطلاق .

٢ - ١٠ رياضيات الحاسب

سكون مقصرين للغاية ما لم نذكر للقارىء بعض المواقف الوخيمة لرياضيات الحاسب .

وحيث أن الحاسب يحتفظ فقط بعدد معين من الأعداد المننوية (بسبب الطول المحدود للكلمة) ، فكل الأرقام داخل الحاسب أرقام حقيقية . وعلى هذا، فنظم الأرقام، بالتحديد كل الأرقام غير الحقيقية مثل $\sqrt{2}$ و π الخ لا تخزن بقيمتها الفعلية ولكن بتقريب حقوى . في واقع الأمر كثير من الأرقام الحقيقية لها عدد محدد من الأرقام المننوية في الصورة الحقيقية إلا أنها لا تنتهى عند التعبير عنها في الصورة الثنائية ، ومن ثم لا تخزن بقيمتها الحقيقية . وتباً لذلك تظهر مشاكل جديدة في رياضيات الحاسب . سنناقش بعضاً منها هنا (أنظر أيضاً المسألة ٢ - ١٧ و ٢ - ١٨) .

(١) خطأ التحويل

تخزن منظم الحاسبات الأرقام في شكلها الثنائى . وبذلك ، فالعدد $1/2$ الذى يمثل في النظام العشري بـ 0.5 يمثل في مثل هذه الحاسبات بـ 0.1 ولكن ، العدد الحقيقى $1/10$ ليس له تمثيل ثنائى محدود .

$$1/10 \dots 0.0001100110011 \dots$$

معتاداً على ملول الكلمة (خلية الذاكرة) يمكن أن يكون خطأ التحويل بسيطاً لا يذكر . ولكن الخطأ سيتوالد كلما تقاسمت عدد العمليات الحسابية .

(ب) توالد خطأ التقريب

على سبيل المثال ، دعنا نفرض أن حاسبنا آلة عشرية تحتفظ بثلاثة خانات معنوية فقط . وبذلك يخزن الرقم $1/3$ في الحاسب بالصورة 0.333 .

اعتبر جمع $1/3$ على نفسها 10 مرات . فنحصل على :

$$\begin{array}{r} .333 \\ + .333 \\ \hline .666 \\ + .333 \\ \hline .999 \\ + .333 \\ \hline 1.332 \end{array}$$

عند هذه النقطة في الجمع ، يحتفظ الحاسب بـ 1.33 فقط حيث أننا افترضنا آلة ذات ثلاث خانات . وإذا أكلنا فنحصل على :

$$\begin{array}{r} 1.33 \\ + .333 \\ \hline 1.66 \\ + .333 \\ \hline 1.99 \\ + .333 \\ \hline 2.32 \\ + .333 \\ \hline 2.65 \\ + .333 \\ \hline 2.98 \\ + .333 \\ \hline 3.31 \end{array}$$

حصلنا على 3.31 بدلا من أن نحصل على القيمة النظرية 3.33 . لقد تولد الخطأ بسرعة إلى حد ما . في الحقيقة ، إذا جمعت 333 . على نفسها 300 مرة في الآلة الخاصة بنا سنحصل على 90.9 فقط بدلا من 99.9 المطلوبة . وعلاوة على ذلك ، إذا أضفت 333 . إلى 100 فسوف يكون الناتج 100 مرة ثانية ، ومهما كانت عدد المرات التي تضيف فيها 333 . إلى 100 ، فسوف تكون النتيجة دائماً 100 وبذلك فإضافة 333 . إلى نفسها بهدورة متتالية سيصل سحبا إلى 100 فقط .

ومن الواضح ، أنه يمكن عمل شيء بخصوص هذه الحالة بالذات ، ولكن سيتولد الأخطاء إن لم تكن حريصين بخصوصها .

(س) عدم توافق (تزامن) الحسابات

أدرس تمييزات الفورتران المكافئة للتمييرات الرياضية .

$$A + (B + C) \quad \text{و} \quad (A + B) + C$$

نفرض أن دقة الحاسب ثمانى خانات معنوية ونفرض أن :

$$A = 0.30000000 \quad B = 87654321. \quad C = -87654322$$

وحيث أن ، الحاسب يحتفظ بثمان خانات فقط ،

$$A + B = 87654321.$$

وحيث أن A لا تؤثر في نتيجة A + B . لذلك :

$$(A + B) + C = 87654321. - 87654322. = -1.0000000$$

ومن ناحية أخرى $B + C = -1.0000000$ ومن م :

$$A + (B + C) = 0.30000000 - 1.0000000 = -0.70000000$$

هناك ٣٠٪ فرق في النتيجة ، وهذه قد تؤدي إلى فارق أكبر إذا استخدمت هذه القيم في حسابات أخرى ، بمعنى أن خطأ التقريب يمكن أن يتزايد .

مسائل محلولة

الـسـوابـت :

٢ - ١ أكتب الآت كـتـوابـت صحـيـحة بالفـورـتـران .

34,275 (أ)	4.8×10^4 (د)	-7.3 (ز)
+275 (ب)	-1,278000 (هـ)	12 (ح)
-23.0 (سـ)	2.137×10^2 (و)	-138 (ط)

34275 (أ)	4800 (د)	- (ز) غير ممكنة - لأنه عدد غير صحيح
+275 (ب)	-1278000 (هـ)	12 (ح)
-23 (سـ)	(و) غير ممكنة - لأنه عدد غير صحيح	-138 (ط)

٢ - ٢ لماذا تعتبر الثوابت الصحيحة الآتية غير مقبولة في الفورتران ؟

2,371 (أ)	-47.0 (ب)	28E3 (سـ)	1234500000 (د)
-----------	-----------	-----------	----------------

(أ) تحتوي على فصلة (و)

(ب) تحتوي على علامة عشرية (.) .

(سـ) العدد الصحيح لا يمكن كتابته في الشكل الأسى .

(د) في معظم الحاسبات ، لا يمكن أن يتكون العدد الصحيح من أكثر من تسعة صفوف .

٢ - ٣ أكتب الآت كـتـوابـت فورـتـران حـقـيـقـية في التـرـانـيـل ' بروح سابقا وفي الشكل الأسى القياسى ،

123 (أ)	5.63×10^8 (د)	0.123 (ز)
12.3 (ب)	-1,234,000 (هـ)	-0.356 (ح)
-3,400 (سـ)	0.47E-10 (و)	3×10^{13} (ط)

0.123 or 0.123E0 (ز)	5.63E-8 or .563E-7 (د)	123. or .123E3 (أ)
-356 or -356E0 (ح)	-1234000. or -.1234E7 (هـ)	12.3 or .123E2 (ب)
3.E13 or .3E14 (ط)	.000347 or .347E-3 (و)	-3400. or -.34E4 (ج)

أنا نؤكد أن الثابت (الحقيقي) يمكن أن يكتب بطرق مختلفة كشكل أسى ولكن بطريقة واحدة فقط في الشكل الأسى القياسى . أى القوة 10 مضروبة في أى رقم يتراوح ما بين 0.1 ، 1.0 ، أو بين 0.1 ، -1.0 .

٢ - لماذا تعتبر الثوابت الحقيقية الآتية غير مقبولة في الفورتران ؟

4E-2 (هـ)	1.6E128 (حـ)	3,248.6 (أ)
4.3E3.7 (و)	E+3 (د)	-13 (ب)

- (أ) يحتوى على فصلة (د)
- (ب) ينقصه العلامة العشرية (ع)
- (ح) لا يمكن أن يكون الأس أكثر من عاشرتين .
- (د) لا يمكن أن يظهر الأس بمفرده (يمكن أن تكتب $3 + 0.0E$) .
- (هـ) يجب أن تحتوى 4 على علامة عشرية .
- (و) معظم الحسابات تسمح فقط بالأس الصحيح .

التفسيرات

٢ - هـ أذكر أى من المتغيرات الصحيحة الآتية غير مقبولة في الفورتران ، ولماذا ؟

2I45 (ز)	J + 329 (هـ)	ALPHA (حـ)	NEXT (أ)
N(3)M (ح)	MACBETH (و)	L124 (د)	FIRST (ب)

- (أ) مقبولة .
- (ب) غير مقبولة - الحرف الأول ليس I أو J أو K أو L أو M أو N .
- (ح) غير مقبولة - الحرف الأول ليس I أو J أو K أو L أو M أو N .
- (د) مقبولة .
- (هـ) غير مقبولة - غير مسموح بعلامة (+) .
- (و) غير مقبولة - أكثر من ستة حروف .
- (ز) غير مقبولة - يجب أن يكون أول حرف أبجدي .
- (حـ) غير مقبولة - يجب أن تكون الحروف أبجدية أو رقمية .

٢ - ٦ أذكر أى من الثوابت الحقيقية الآتية غير مقبولة ، ولماذا ؟

STOP (ز)	ALTITUDE (هـ)	ROOT2 (حـ)	ANSWER (أ)
X-RAY (ح)	4XYZ (و)	MAX (د)	LAMBDA (ب)

- (أ) مقبولة .
 (ب) غير مقبولة - الحرف الأول L محجوز للمتغيرات الصحيحة .
 (ج) مقبولة .
 (د) غير مقبولة - الحرف الأول M محجوز للمتغيرات الصحيحة .
 (هـ) غير مقبولة - يجب أن يكون أول حرف أبجدياً .
 (ز) غير مقبولة - لأنها كلمة دالة من كلمات الفورتران .
 (ح) غير مقبولة - الشرطة (-) ليست حرفاً أو رقماً .

التعبيرات الآتية والعمليات :

٢ - ٧ بفرض أن التعبيرات ذات النمط المختلط غير مقبولة لدى الحاسب ، حدد أيًا من التعبيرات الآتية مقبولة كتعبيرات فورتران وأوجد قيمة كل منها .

(ز) $0 + 4.6$	(أ) $3 * + 5$	(ح) $3 * 4.5$	(أ) $2.4 + 3.85$
(ح) $-7/3$	(و) $4.3 - 7.93$	(د) $3. * * 2$	(ب) $12/5$

(أ) 6.25

- (ب) 2 (تم بتر الكسر في القسمة الصحيحة) .
 (ح) غير مقبول - لأن حدود التعبير من نوعين مختلفين .
 (د) 9 (يمكن أن يكون الأس عدد صحيحاً بالرغم من أن الأساس حقيقي) .
 (هـ) غير مقبولة - لا يمكن أن تظهر العلامتين * و + متتاليتين .
 (و) $3.63 \dots$
 (ز) غير مقبولة - لأن حدود التعبير من نوعين مختلفين .
 (ح) 2 - (تم بتر الكسر في القسمة الصحيحة) .

٢ - ٨ نفرض أن حسابات النمط المختلط مقبولة ، أوجد قيمة كل من التعبيرات التالية :

(ز) $2 * (3 * * 2)$	(أ) $7 / (4 * 2)$	(ج) $2 * 6 / 5$	(أ) $3 + 4.8 * 2$
(ح) $2 * 3 * * 2$	(و) $7 / (4 * 2)$	(د) $2 * 6 / 5$	(ب) $5.2 + 12 / 8$

(أ) للضرب أسبقية على الجمع ، وبذلك :

$4.8 * 2$ تعطي 9.6 و $3 + 9.6$ تعطي 12.6

- (ب) القسمة الصحيحة $12/8$ تعطي 1 و 1 + 5.2 تعطي 6.2 .
 (ح) $2 * 6$ تعطي الرقم الحقيقي 12 و $12 / 5$ تعطي الرقم الحقيقي 2.4 .
 (د) $2 * 6$ تعطي العدد الصحيح 12 والقسمة الصحيحة $12/5$ تعطي 2 وتكون هي النتيجة
 (هـ) $4 * 2$ تعطي الرقم الحقيقي 8 و $7/8$ تعطي 0.875
 (و) $2 * 4$ تعطي الرقم الصحيح 8 ، والقسمة الصحيحة $7/2$ تعطي 0 وتكون هي النتيجة

18 (ز)

(ح) الرفع إلى الأس له أسبقية على الضرب ، ومن ثم تحصل على 18 كما في (ز)

٢-٩ أكتب تعبير فورتران مناظر لكل من التعبيرات الرياضية التالية :

$$\frac{x^4}{4!} \quad (و) \quad a + \frac{b}{c^2} \quad (د) \quad (x + y)(u + v) \quad (أ)$$

$$3(x + y) \quad (ب)$$

$$\frac{a + b}{c \cdot d} \quad (هـ) \quad 3xy^2 - 2x^2y \quad (ج)$$

$$(A + B)/(C * D) \quad (هـ) \quad 3.0 * X * Y ** 2 - 2.0 * X ** 2 * Y \quad (ج) \quad (X + Y) * (U + V) \quad (أ)$$

$$X ** 4 / (4. * 3. * 2. * 1.) \quad (و) \quad A + B / C ** 2 \quad (د) \quad 3.0 * (X + Y) \quad (ب)$$

٢-١٠ أوجد تعبيراً مكافئاً لكل من تعبيرات الفورتران التالية ، أي التعبير الذي لا يغير ترتيب العمليات الحسابية ، وذلك بحذف الأقواس الزائدة .

$$(A * B * C) / ((X * Y) ** 2) \quad (ج) \quad (A * B) * (C + D) \quad (أ)$$

$$A * (B * C) \quad (د) \quad (A * (B ** 2)) / (C * D) \quad (ب)$$

$$A * B * (C + D) \quad (أ)$$

$$A * B ** 2 / (C * D) \quad (ب)$$

$$A * B * C / (X * Y) ** 2 \quad (ج)$$

(د) $A * (B * C)$ لا يمكن حذف الأقواس رغم أن $A * B * C$ ، تحسب قيمتها بالصورة $(A * B) * C$ ، تكافئ رياضياً $A * (B * C)$ ، إلا أن ترتيب العمليات مختلف وبذلك قد يؤدي خطأ التقريب إلى نتائج مختلفة . أنظر قسم ٢-١٠ .

٢-١١ أكتب تعبير فورتران مناظراً لكل من التعبيرات الرياضية التالية وذلك باستخدام اللوال الرياضية في جدول ٢-٢ (صفحة ٣٥)

$$\frac{1}{|a \cdot b|} + c \quad (ج) \quad \sqrt{a^2 + b^2} \quad (أ)$$

$$\cos(\log_{10}(a + 3b)) \quad (د) \quad e^{x+y} - \sin(x + ny) \quad (ب)$$

$$\text{SQRT}(A ** 2 + B ** 2) \quad (أ)$$

$$\text{EXP}(X + Y) - \text{SIN}(X + N * Y) \quad (ب)$$

إذا كان الحاسب لا يقبل تعبيرات ذات نمط مختلط

$$\text{EXP}(X + Y) - \text{SIN}(X + \text{FLOAT}(N) * Y)$$

$$1. / \text{ABS}(A * B) + C \quad (ج)$$

$$\text{COS}(\text{ALOG}_{10}(A + 3.0 * B)) \quad (د)$$

٢-١٢ فيما يلي تعبيرات رياضية تناظرها تعبيرات فورتران غير صحيحة . أكتب تعبيرات الفورتران الصحيحة :

$$\left(\frac{a}{b+c}\right)^2, A/(B+C)**2 \quad (-) \quad \frac{a \cdot b}{c \cdot d \cdot e}, AB/CDE \quad (أ)$$

$$\sqrt{\frac{a^2}{b+c}}, \text{SQRT}(A**2/(B+C)) \quad (د) \quad \left(\frac{x}{y}\right)^{n+1}, (X/Y)**N+1 \quad (ب)$$

$$\text{SQRT}(A**2/(B+C)) \quad (د) \quad (A/(B+C))**2 \quad (-) \quad (X/Y)**(N+1) \quad (ب) \quad A*B/(C*D*E) \quad (أ)$$

الجمل الحسابية

٢-١٣ إدرس الجمل الآتية :

$$AB = CD \quad (أ) \quad A = A + A \quad (-) \quad A = B + C \quad (أ)$$

$$A = ABS(A) \quad (و) \quad A + B = C + D \quad (د) \quad B + C = A \quad (ب)$$

أى من الجمل السابقة مقبول كجمل فورتران ؟ وماهى الشروط الضمنية التى افترضناها لتلك الجمل المقبولة ؟ تذكر أن المتغير ، أى اسم مكان التخزين فقط ، هو الذى يمكن أن يظهر على يسار علامة التساوى .

(أ) مقبولة .

(ب) غير مقبولة حيث $B + C$ ليست متغيراً .

(ج) مقبولة .

(د) غير مقبولة حيث $A + B$ ليست متغيراً .

(هـ) مقبولة .

(و) مقبولة .

نحن نفترض ضمناً أن كل المتغيرات على الجانب الأيمن من علامة التساوى قد تم تعريفها مسبقاً ، أى أن الأرقام موجودة فعلاً فى أماكن التخزين التى تحمل هذه الأسماء .

٢-١٤ نفرض أن A و B و J و K تحتوى على القيم الآتية : $A = 2.7$ و $B = 3.5$ و $J = 3$ و $K = 2$. أوجد قيم X و L بعد كل زوج من الجمل التالية :

$$X = K/3*A/2 \quad (-) \quad X = A + J*K**2 + B \quad (أ)$$

$$L = K/3*A/2 \quad L = A + J*K**2 + B$$

$$X = ABS(A - J*B)/5 \quad (د) \quad X = 5*J/4*K \quad (ب)$$

$$L = ABS(A - J*B)/5 \quad L = 5*J/4.0*K$$

(أ) عند إيجاد قيمة التعبير يقوم الحاسب بتنفيذ رفع الأس أولاً ، ثم بعد ذلك الضرب وأخيراً الجمع .

$$A + J*K**2 + B = 2.7 + 3(-2)^2 + 3.5 = 2.7 + 12 + 3.5 = 18.2$$

يخزن الحاسب 18.2 فى X حيث أن X متغير حقيقى ، ولكن حيث أن L متغير صحيح ، فيخزن الحاسب 18 و L (أى ، المرء الصحيح فقط من 18.2) أى أن $X = 18.2$ و $L = 18$

(ب) ينفذ الحاسب الضرب والنسبة من اليسار إلى اليمين :

$$5*J/4*K = 15/4*K = 3*K = 6$$

$$5*J/4.0*K = 15/4.0*K = 3.75*K = -7.5$$

وبسبب القسمة الصحيحة ، فإن $15/4$ تعطي 3 ولكن $15/4.0$ تعطي 3.75 وحيث أن X حقيقية ، فتحول 6 — إلى رقم حقيقي 6.0 — ثم تخزن بعد ذلك في X وحيث أن L متغير صحيح ، فقد تم بت 7.5 — وأصبحت 7 — ثم بعد ذلك تم تخزينها في L . وبذلك أصبحت $X = -6.0$ و $L = -7$

(ح) بسبب القسمة الصحيحة ، فإن $K/3$ تعطي 0 ومن ثم $X = 0.0$ و $L = 0$

(د) $ABS(A - J*B)/5 = |2.7 - 3(3.5)|/5 = |-7.8|/5 = 1.56$. So $X = 1.56$ and $L = 1$

٢-١٥ افترض أن A و B لهما القيم التالية $A = 2.5$ و $B = 3.5$ أوجد قيم A و B بعد تنفيذ كل من مجموعات الجمل التالية :

$$T = A \quad (\text{ب})$$

$$A = B \quad (\text{أ})$$

$$A = B$$

$$B = A$$

$$B = T$$

(أ) توجه الجملة $B = A$ الحاسب بأن يسمح القيمة الحالية الموجودة في A ويموض عنها بالقيمة الحالية لـ B ومن ثم .

$$A \leftarrow 3.5$$

وتوجه الجملة التالية $B = A$ الحاسب بأن يسمح القيمة الحالية لـ B (ويموض) عنها بالقيمة الحالية لـ A ومن ثم .

$$B \leftarrow 3.5$$

لاحظ أن قيمة B لم تتغير .

(ب) تعطي الجمل الثلاث .

$$T \leftarrow 2.5, \quad A \leftarrow 3.5, \quad B \leftarrow 2.5$$

لاحظ أن قيم A و B قد تم تبديلها عن طريق تقديم مكان تخزين مساعد ليحتفظ بقيمة أحد المتغيرات خلال عملية التبديل .

٢-١٦ افترض أن المتغيرات A و B و C قد تم تعريفها مسبقاً . اكتب جملة الفورتران التي (أ) تضاعف قيمة A (ب)

تزيد قيمة B بأربعة (ج) تقلل قيمة C بالقيمة الموجودة في A (د) تخزن القيمة المتوسطة لـ A و B و C

في AVE أي مجموع A و B و C مقسوماً على 3. (هـ) تخزن الجذر التربيعي لمجموع مربعات A و B و C في DIST

$$\text{أي } \sqrt{A^2 + B^2 + C^2}$$

$$\text{DIST} = \text{SQRT}(A**2 + B**2 + C**2) \quad (\text{هـ}) \quad C = C - A \quad (\text{ج}) \quad A = 2.0 * A \quad (\text{أ})$$

$$\text{AVE} = (A + B + C)/3.0 \quad (\text{د}) \quad B = B + 4.0 \quad (\text{ب})$$

مسائل متنوعة :

٢-١٧ ناقش الطرق الثلاث الآتية لكتابة x^2 في الفورتران :

$$X**2.0 \quad (\text{ج})$$

$$X**2 \quad (\text{ب})$$

$$X*X \quad (\text{أ})$$

في (أ) تكتب x^2 بدلالة الضرب وسوف يتم حساب قيمتها بهذه الطريقة في (ب) ستحسب قيمة $X**2$ بالضرب

كنتيجة لترجمة $X**2$ إلى $X*X$ في (ج) الأس حقيقى وبذلك فإن (ج) ستحسب قيمتها $\text{EXP}(2.0 | \text{ALOG}(X))$

بشرط أن تكون X موجبة . (انظر قسم ٢-٨) ويتضح أن (أ) و (ب) أفضل من (ج) . في الحقيقة يمكن أن تعطي

(ج) نتيجة مختلفة نسبياً في غنوه خطأ التقريب عند حساب قيمة EXP و ALOG وأكثر من ذلك ، فبالرغم أن (أ)

و (ب) تتطلب نفس وقت التشغيل ، فإن (أ) لا تتطلب ترجمة إضافية .

٢-١٨ ناقش جزئى البرنامج التاليين :

$$\text{AVE} = \text{SUM}/\text{FLOAT}(N) \quad (\text{ب}) \quad \text{XN} = N \quad (\text{أ})$$

$$\text{AVE} = \text{SUM}/\text{XN}$$

(تنبة أن كليهما مكتوب لتجنب قسمة النمط المختلط)

في (١) تحول الرقم الصحيح N أولاً إلى رقم حقيقي وبنزن في XN ثم نحسب المتوسط SUM/XN لاحظ أن ذلك يتطلب مكان تخزين إضافي . ومن وجهة أخرى ، فإن (ب) تتطلب جملة واحدة حيث أننا نستخدم الدالة FLOAT في الحسابات . وهكذا . فتبدو (ب) أكثر فائدة من (١) . ولكن ، إذا كان البرنامج طويلاً ، ونحتاج إلى تحويل N إلى رقم حقيقي عدة مرات ، لكان من الأحسن تحويل N مرة واحدة إلى رقم حقيقي بواسطة N = XN بدلاً من استخدام FLOAT (N) طوال البرنامج .

مسائل تكميلية

الثوابت :

٢-١٩ اكتب ما يلي في صورة ثوابت فورتران صحيحة .

23.51 (ز)	3.1214×10^3 (د)	2,348 (ا)
1,250,000 (ح)	-37.0 (هـ)	5.31×10^3 (بـ)
$57,000.0 \times 10^{-2}$ (ط)	$21,500 \times 10^{-3}$ (و)	-531 (...)

٢-٢٠ لماذا تعتبر ثوابت الفورتران الصحيحة التالية غير مقبولة ؟

٣٨٤٠٠٠٠٠٠٠٠ (د)	37810000000 (ج)	-784.0 (ب)	2,578 (ا)
-------------------	-------------------	--------------	-------------

٢-٢١ اكتب القيم التالية في شكل ثوابت فورتران حقيقية وأيضاً في الشكل الأسى القياسى .

.00005829 (هـ)	2,348,500 (ج)	2,345 (ا)
21×10^{100} (و)	-7.63×10^{-5} (د)	-1.63 (ب)

٢-٢٢ لماذا تعتبر ما يلي ثوابت فورتران حقيقية غير مقبولة ؟

E21 (هـ)	52E-7 (ج)	-3.6E134 (ا)
-1,378.0 (و)	256 (د)	2,356 4 (ب)

٢-٢٣ حدد أى زوج من الثوابت يمثل نفس العدد .

12345 (د)	+12345 (د)	43.6 (ا)	4.36E01 (ا)
0.234E6 (هـ)	234.E4 (هـ)	543 (ب)	5.43E+02 (ب)
1.00 (و)	1. (و)	0.00004 (ج)	4.0E-5 (ج)

المتغيرات :

٢-٢٤ ادرس قائمة الأسماء التالية . وحدد من بينها ما هو مقبول أما (١) كتغيرات صحيحة أو (٢) كتغيرات سقذبة . اذكر السبب في أن بقية أسماء المتغيرات غير مقبولة ؟

AMOUNT (ك)	A567B (و)	MORE (ا)
A1567 (ل)	RATE (ز)	LESS (ب)
A1245 (م)	X34.7 (ح)	NEITHER (جـ)
IN.OUT (ن)	GAMMA (ط)	AAAAAA (د)
OPERAND (ص)	KAPPA (ي)	SPQR (هـ)

العمليات والمتغيرات الرياضية

٢ - ٢٥ بفرض أن الحاسب لا يقبل حساب النمط المختلط . حدد مما يلي ما هو مقبول كتعبيرات فورتران وأوجد قيمته . عرف وحدد الأخطاء في التعبيرات غير المقبولة .

9/-4 (ز)	0/3.4 (هـ)	6.*4 (ج)	6.3 + 5.2E3 (ا)
-4/9 (ح)	6*-7 (و)	6.**4 (د)	-18/7 (ب)

٢ - ٢٦ افرض أن الحاسب يقبل فعلا حساب النمط المختلط . أوجد قيمة كل تعبير مما يلي :

'7/3)/(6/5) (ز)	-3*2**3 (هـ)	19/(2*5) (ج)	3 - 5*2.5 (ا)
2**2*2/3 (ح)	-3*(2**3) (و)	19/(2.*5) (د)	2.8 - 17/5 (ب)

٢ - ٢٧ اكتب تعبير فورتران مناظراً لكل تعبير رياضي مما يلي :

$(3 + \frac{a}{b})^{m-1}$ (هـ)	$(2x + y)(3z - 4w)$ (ا)
$\frac{a}{b} + 6$ (و)	$4x^2y - 3xy + 7yz^3$ (ب)
$x - \frac{y}{z}$ (ز)	$(\frac{a+b}{c+d})^3$ (ج)
	$\frac{x^5}{5!}$ (د)

٢ - ٢٨ أوجد التعبير المكافئ لكل تعبير من تعبيرات الفورتران الآتية . أى التعبير الذى لا يغير ترتيب العمليات الحسابية بحذف الأقواس غير اللازمة (الزائدة) .

$(X*(Y-Z))*(A**2)$ (هـ)	$A + ((B*C)/D)$ (ج)	$(X + Y) + Z$ (ا)
$((A**2) + (B**2)) - (D*(E/F))$ (و)	$(A + (B**3))/(X*Y)$ (د)	$X + (Y + Z)$ (ب)

٢ - ٢٩ اكتب تعبيرات الفورتران المناظرة لكل من التعبيرات الرياضية التالية باستخدام الدوال الرياضية الموضحة في جدول ٢ - ٢٠ (صفحة ٣٥) .

$\sqrt{5x^2 + 8y^2}$ (هـ)	$\log_e(x + y)^2$ (ا)
$\sin(x - 2y) + e^{xy} - x^2 - y^2 $ (و)	$\log_{10}(a \cdot b)^2$ (ب)
$e^{ a } - \frac{b^2}{ c }$ (د)	$ \sqrt{x^2 + y^2} - \cos(a + b) $ (ج)
$\sqrt{ \cos(a - nb) }$ (ح)	$\sqrt{ \sin(a - b) }$ (د)

١ - ٣٠ فيما يلي تعبيرات رياضية وتعبيرات فورتران غير صحيحة . اكتب تعبيرات الفورتران الصحيحة .

$a + \frac{b}{c \cdot d}$ (ا)	$A + B/CD$ (ا)
$\frac{x^{n+1}}{y^{n-1}}$ (ب)	$(X**N + 1)/(Y**N - 1)$ (ب)
$\sin(x + n)$ (ج)	$SIN(X + FLOAT(N))$ (ج)
$\log_{10} a \cdot b $ (د)	$LOG(ABS(AB))$ (د)

اجمل الحاسبة

٢ - ٢١ أى من جمل الفورتران الآتية مقبولة ؟

$$\begin{aligned} ABC &= DEF & (ز) \\ ABC &= 2DEF & (ح) \\ A*B+C &= D*E*F & (ط) \end{aligned}$$

$$\begin{aligned} XY &= ZW & (د) \\ XY &= Z*W & (هـ) \\ X*Y &= Z*W & (و) \end{aligned}$$

$$\begin{aligned} X &= Y + Z & (ا) \\ X + Y &= Z & (ب) \\ Y + Z &= X & (ج) \end{aligned}$$

٢ - ٢٢ اعتبر أن X و Y و L و M تحتوى على القيم التالية $X = 3.1$ و $Y = 4.6$ و $L = 2$ و $M = 3$ أوجد القيمة النهائية لكل من A و J بعد تنفيذ مجموعات الجمل التالية :

$$\begin{aligned} A &= X + 2*Y & (د) \\ A &= 2*A + 4 \\ J &= A \end{aligned}$$

$$\begin{aligned} A &= X - 2*Y + X/L**2 & (ا) \\ J &= X - 2*Y + (X/L)**2 \end{aligned}$$

$$\begin{aligned} J &= L + 3*M & (هـ) \\ J &= J**2 + Y \\ A &= X + J \end{aligned}$$

$$\begin{aligned} A &= 8/M + 3*L & (ب) \\ J &= 8.0/M - Y \end{aligned}$$

$$\begin{aligned} J &= \text{SQRT}(X*Y) & (و) \\ A &= J*Y \\ J &= A*M \\ A &= A + \text{ABS}(\text{FLOAT}(J)) \end{aligned}$$

$$\begin{aligned} A &= L/M*X/Y & (ج) \\ J &= L/(M + X)*Y \end{aligned}$$

٢ - ٢٣ اعتبر أن X و Y و Z قد تم تعريفها مسبقاً . اكتب الجملة الحاسوبية التي تنجز الآت : (أ) تزيد قيمة X ب 3.2 (ب) تجعل قيمة Y ثلاثة أضعاف ، (ج) تربيع قيمة Z (د) تخزن حاصل ضرب القيم X و Y و Z في PRDT (هـ) تخزن طول الوتر لثلث قائم الزاوية في HYP وأضلاعه لها الأطوال X و Y (و) تخزن القيمة المتوسطة لكل من X و Y و Z في AVE

٢ - ٢٤ افرض أن A و B و C قد تم تعريفها مسبقاً . اكتب جزء البرنامج الذي يبديل قيم A و B و C بحيث تأخذ A قيمة B وتأخذ B قيمة C وتأخذ C قيمة A (تليخ : انظر مسألة ٢ - ١٥ (ب)) .

اجابات للمسائل التكميلية المختارة

١٩ - ٢ (أ) 234^٧ (ب) 5310 (ج) 531 - (د) مستحيل (هـ) 37 -- (و) مستحيل
(ز) مستحيل (ح) 1250000 (ط) 570

٢٠ - ٢ (أ) تحتوي على فصلة (٠)

(ب) تحتوي على علامة عشرية (٠)

(ج) لا يمكن أن يكون العدد الصحيح في معظم الحسابات أكثر من تسعة حروف .

(د) لا يمكن أن يكتب العدد الصحيح في الشكل الأسّي .

٢١ - ٢ (أ) 2345. أو 0.2345E4 (ب) -1.63 أو -0.163E1 (ج) 2348500. أو 0.23485E7
(د) -0.763E-4 أو -7.63E-5 (هـ) 0.5829E-4 أو 0.00005829 (و) 0.21E20 أو 21.E18

٢٢ - ٢ (أ) لا يمكن أن يكون الأس أكثر من رقمين في معظم الحسابات .

(ب) يحتوي على فصلة (و) .

(ج) تنقصه العلامة العشرية (٠) .

(د) تنقصه العلامة العشرية (٠) .

(هـ) لا يمكن أن يظهر الأس بمفرده . (يمكن أن يكتب 0.0E21)

(و) يحتوي على فصلة (:) .

٢٣ - ٢ (أ) نعم . (ب) ليس أي منهما عدداً صحيحاً . (ج) نعم . (د) نعم . (هـ) لا . (و) نعم .

٢٤ - ٢ (أ) عدد صحيح (ب) عدد صحيح . (ج) أكثر من ستة حروف (د) حقيق
(هـ) يبدأ برقم (و) حقيق (ز) حقيق (ح) غير مسموح بالعلامة العشرية
(ط) حقيق (ي) عدد صحيح (ك) حقيقية (ل) غير مسموح بعلامة +
(م) حقيق (ن) غير مسموح بالشرطة (-) (س) أكثر من ستة حروف .

٢٥ - ٢ (أ) 5206.3 (ب) 2 -- (ج) غير مقبول (د) 1296. (هـ) غير مقبول . (و) غير مقبول .
حيث لا يمكن أن تظهر ٥ و -- متتاليين
(ز) 2 - (ح) 0

٢٦ - ٢ (أ) 9.5 -- (ب) 0.2 -- (ج) 1 (د) 1.9 (هـ) 24 - (و) 24 -- (ز) 2 (ح) 2

٢٧ - ٢ (أ) $(2.*X + Y)*(3.*Z - 4.*W)$ (ب) $4.*X**2*Y - 3.*X*Y + 7.*Y*Z**3$ (ج) $((A + B)/(C + D))**3$
(د) $X**5/(5.*4.*3.*2.*1.)$ (هـ) $(3. + A/B)**(M - 1)$ (و) $(A/B + 6.)/(X - Y/Z)$

الفصل الثاني : جمل رياضية

•.

$(A + B**3)/(X*Y)$	(د)	$X + Y + Z$	(أ) ٢٨ - ٢
$X*(Y - Z)*A**2$	(أ)	$X + (Y + Z)$	(ب)
$A**2 + B**2 - D*(E/F)$	(و)	$A + B*C/D$	(ج)

$SQRT(5.*X**2 + 8.*Y**2)$	(أ) ٢٩ - ٢
$SIN(X - 2.*Y) + EXP(X*Y) - ABS(X**2 - Y**2)$	(ب)
$EXP(ABS(A)) - B**2/ABS(C)$	(ج)
$SQRT(ABS(COS(A - FLOAT(N)*B)))$	(د)
$ALOG((X + Y)**2)$	(أ)
$ALOG10((A - B)**2)$	(و)
$ABS(SQRT(X - Y**3) - Z**3/COS(A + B))$	(ز)
$SQRT(ABS(SIN(A - ABS(B))))$	(ح)

$SIN(X + FLOAT(N))$	(ج)	$A + B/(C*D)$	(أ) ٣٠ - ٢
$ALOG10(ABS(A*B))$	(د)	$(X**(N + 1))/(Y**(N - 1))$ or $X**(N + 1)/Y**(N - 1)$	(ب)

٣١ - ٢ (أ) نم . (ب) لا . (ج) لا . (د) نم . (أ) نم . (و) لا . (ز) نم . (ج) لا . (ط) لا .

$A = 56.1, J = 53$	(أ)	$A = 0.0, J = 92$	(ج)	$A = -5.325, J = -3$	(أ) ٣٢ - ٢
$A = 54.8, J = -41$	(و)	$A = 28.6, J = 28$	(د)	$A = 4.0, J = -7$	(ب)

$X = X + 3.2$	(أ) ٣٣ - ٢
$Y = 3.*Y$	(ب)
$Z = Z**2$	(ج)
$PRDT = X*Y*Z$	(د)
$HYP = SQRT(X**2 + Y**2)$	(أ)
$AVE = (X + Y + Z)/3.0$	(و)

$T = A$	٣٤ - ٢
$A = B$	
$B = C$	
$C = T$	

الفصل الثالث

الإدخال / الإخراج العددي

٣ - ١ مقدمة

سنرى في هذا الفصل . كيف يقرأ الحاسب البيانات الخارجية (المدخلات) ويطبع البيانات (المخرجات) يتم هذا عن طريق جملتي READ و WRITE على الترتيب . وتسمى هذه عمليات الإدخال / الإخراج أو ببساطة عمليات (I/O) . وكما ذكرنا مسبقاً فسنعتبر أن وحدة الإدخال هي قارئ البطاقات ، ووحدة الإخراج هي آلة الطباعة (إلا إذا نصر أو تضمن على غير ذلك) .

تأمر جملة READ الحاسب بأن يقرأ معلومات عديدة أو رمزية من مجموعة بطاقات (مجموعة بطاقات البيانات) . وتأمر جملة WRITE الحاسب بأن يطبع معلومات على آلة الطباعة . عادة ما تصاحب جملة READ أو WRITE جملة FORMAT . في حالة الإدخال فإن جملة FORMAT تمد الحاسب بنوع المعلومات والأماكن الخاصة بها في سجل إدخال . في حالة الإخراج ، نجد أن جملة FORMAT تمد الحاسب أيضاً بإمكان طباعة المعلومات . ومن الواضح أن جملة FORMAT يمكن أن تكون مفصلة جداً . ولكن كثيراً من الحاسبات تسمح بما يسمى ملامح الإدخال - الإخراج غير المصاغ . سنناقش أولاً مثل هذا (I/O) غير المصاغ . (تستخدم بعض الكتب المصطلح « صياغة غير مقيدة » فضلاً عن المصطلح « غير مصاغ ») .

٣ - ٢ إدخال / إخراج غير مصاغ

بعد قراءة هذا القسم ، يمكن للقارئ أن يمضي إلى قسم ٣ - ١٠ ، والفصول الثلاثة التالية قبل العودة إلى بقية هذا الفصل (إلا إذا كانت إمكانيات الحاسب المتاح للقارئ لا تسمح بإدخال - إخراج غير مصاغ) .

ويمكن أن يختلف (I/O) غير المصاغ ، والذي يتم مناقشته في هذا الفصل ، من جهاز إلى آخر باختلافات طفيفة . ولذا ، ننصح القارئ بالرجوع إلى التفصيلات الخاصة بالحاسب المتاح له .

فيما يلي جملة PRINT غير المصاغة النموذجية :

PRINT, A, B, C, AREA, M, N

لاحظ أن كلمة PRINT هي وكل متغير فيما عدا المتغير الأخير متنوعة بفصلة . تأمر هذه الجملة الحاسب بأن يطبع قيم المتغيرات A ، B ، C ، AREA ، M ، N بهذا الترتيب المعطى . ستطبع القيم بشكل مخصص (مروف) مسبقاً ، وسوف تطبع الأرقام الحقيقية عادة في الشكل الأسّي . يجب تحديد عدد المتغيرات التي يمكن أن تطبع في السطر الواحد أيضاً مسبقاً . بينما سيبدأ الحاسب الطباعة على سطر جديد كلما قابل جملة PRINT .

فيما يلي جملة READ غير المصاغة النموذجية :

READ, A, B, C, LOT, AREA

لاحظ أن كلمة READ وكل متغير فيها عدا الأخير متبوعة بفصلة . تأمر هذه الجملة الحاسب بأن « يقرأ » خمسة أرقام . رقم لكل متغير (مبتدأ) من أول بطاقة في مجموعة البيانات . إذا كان هناك خمسة أرقام أو أكثر على هذه البطاقة ، فسوف تخصص أول خمس قيم إلى A و B و C و LOT و AREA على الترتيب ، وتهمل البطاقة بعد ذلك . ولكن إذا كان هناك أقل من خمسة أرقام على البطاقة فسببوصول الحاسب القراءة من مجموعة البيانات إلى أن يجد الأرقام الصحيحة جميعها . أي أن ، تنفيذ جملة READ لا يكتمل إلا إذا خصصت قيم لكل المتغيرات المذكورة في القائمة .

لاحظ أن بطاقة البيانات تهمل بعد أن تستخدم في القراءة . ومن ثم تكون دائماً البطاقة التي لم تقرأ بعد على قة مجموعة البيانات .

يجب أن تفعل الأرقام التي تقرأ من مجموعة بطاقات البيانات بجملة إدخال غير مصاغة بفصلات (تسمح بعض مراكز الحاسبات بأن تكون الأرقام مفصولة بواسطة مسافة فارغة (خالية) بدلا من الفصلة) . تثقب الأرقام الحقيقية أما في الشكل العشري أو في الشكل الأسّي ومن الواضح انه يجب أن تكون الأرقام الموجودة على البطاقة (بطاقات) هي والمتغيرات المناظرة لها من نفس النوع .

مثال ٣ - ١

ا. أن حاسبا ينفذ الجملتين التاليتين :

READ, I, J, K
READ, L, M

واعتبر أن بطاقات البيانات قد تم تثقيبها كالتالي :

222,76	: أول بطاقة بيانات
38,175,55	: ثاني بطاقة بيانات
194,58,567	: ثالث بطاقة بيانات

حين ينفذ الحاسب جملة READ الأولى يخصص 222 و 76 إلى I و J على الترتيب وبعد ذلك يذهب إلى بطاقة البيانات الثانية . ويخصص 38 إلى K ويكمل هذا تنفيذ جملة READ الأولى وتهمل بطاقة البيانات الثانية . رغم أن الأرقام 175 و 55 لم تقرأ . حين ينفذ الحاسب جملة READ الثانية ، سيقرا البطاقة التي على قة مجموعة بطاقات البيانات . حينئذ يستخدم الحاسب بطاقة البيانات الثالثة لتخصيص 194 إلى L و 58 إلى M وعلى ذلك يكون لدينا

$$I = 222, J = 76, K = 38, L = 194, M = 58$$

تتطلب بعض التسهيلات الحاسوبية أن تكون الأرقام على بطاقات البيانات مفصولة بواسطة مسافات خالية (يرمز إليها بـ b كدليل) بدلا من الفصلات (,) . في هذه الحالات يجب أن تثقب البيانات كالتالي :

222 _b 76	: أول بطاقة بيانات
38 _b 175 _b 55	: ثاني بطاقة بيانات
194 _b 58 _b 567 _b	: ثالث بطاقة بيانات

٢ - ٢ مقدمة للإدخال / الإخراج المصاغ

تم تقديم جمل الإدخال - الإخراج غير المصاغ (الصياغة غير المقيدة) أولا ، حتى يتمكن القراء من البدء في كتابة البرامج والتعامل مع الحاسب في أقرب وقت ممكن . وبدا لتجنب التعابير حتى نلم بتفاصيل كتابة جمل FORMAT . ومع ذلك ، فلأن I/O غير

الفصل الثالث : الإدخال / الإخراج العددي

٧٠

المصاغ تنفذ فعلا بصيغ سبق تحديدها ، فلا نملك تحكما كبيرا في كيفية قراءة وطباعة المعلومات . نناقش هنا تفاصيل كتابة جمل I/O المصاغة وستنصير مناقشتنا على I/O الخاص بالقيم الرقمية . وسنعالج الأنواع الأخرى في فصل لاحق .

في جهاز حاسب كبير يمكن استخدام وحدات كبيرة للإدخال والإخراج . وللترقية بين هذه الوحدات ، يخصص لهم أرقام - يوضح جدول ٣ - ١ أرقام وحدات نمطية .

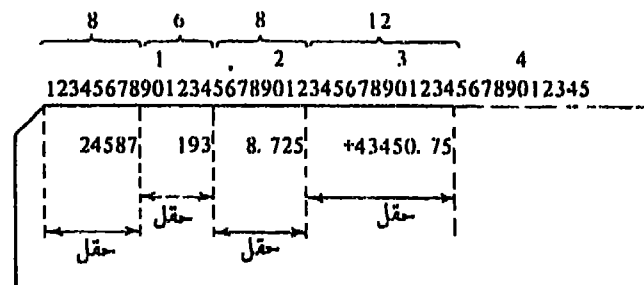
جدول ٣ - ١

الرقم	للوحدة
١	وحدة الشرائط
٥	قارئ البطاقات
٦	طابعة سطرية
٧	مقثب البطاقات

(تم استخدام هذه الأرقام أولا في كثير من حاسبات IBM وما تزال منتشرة الاستخدام) في هذا الكتاب ، نفترض أن وحدة الإدخال الخاصة بنا وهي قارئ البطاقات ، وتحمل رقم 5 ووحدة الإخراج الخاصة بنا وهي آلة الطباعة وتحمل رقم 6 .

حيث أن ، وحدة الإدخال هي قارئ البطاقات ، فستعطي بيانات الإدخال إلى الحاسب عن طريق مجموعة بطاقات . وهي تشبه البطاقات التي تقثب عليها جمل الفورتران تماما . إلا أن قوانين تثقيب جمل الفورتران لا تنطبق هنا ، أي ، يمكن أن نستخدم التمايز عموداً بالكامل من بطاقة البيانات ، وليس لأي عمود غرض معين كما في عمود 6 مثلا .

تقثب المدخلات في مجموعة من الأعمدة المتلاصقة على بطاقة البيانات . يوضح شكل ٣ - ١ بطاقة بيانات مقثبة . عاينها أرقام ، الرقم الأول 24587 مقثب في الأعمدة من 1 إلى 8 الرقم الثاني 193 مقثب في الأعمدة من 9 إلى 14 الرقم الثالث 8.725 مقثب في الأعمدة من 15 إلى 22 والرقم الرابع 43450.75 - مقثب في الأعمدة من 23 إلى 34 وتسمى مجموعة الأعمدة المتلاصقة هذه حقول ، ويسمى عدد الأعمدة في الحقل بعرض الحقل . لاحظ أن كل رقم مقثب في حقله مضبوط - جهة اليمين (right-justified) أي تظهر آخر خانة من الرقم في العمود الأخير من حقله . (تستخدم بعض المصطلح منظم - يميني بدلا من المصطلح مضبوط جهة اليمين).



شكل ٣ - ١ بطاقة بيانات توضح الحقول وعرض الحقول .

افترض أن الأرقام الأربعة السابقة ستخصص للمتغيرات ID و LOT و RATE و PRICE على الترتيب . يتم هذا في الفورتران بواسطة جملة READ التي تصاحبها جملة FORMAT والتي قد تكتب كالتالي :

READ(5, 100) ID, LOT, RATE, PRICE
100 FORMAT(I8, I6, F8.0, F12.0)

بدراسة الرقبن بن القوسين اللذين يتبعان كلمة READ نجد أن الرقم الأول 5 هو رقم وحدة الإدخال يشير إلى قارئ البطاقات (كنا سنكتب 1 بدلا من 5 إذا أردنا استخدام وحدة الشرائط بدلا من قارئ البطاقات ، إذا كانت بيانات الإدخال مخزنة على شرائط مغنطة مثلا) ونجد أن الرقم الثاني 100 يشير إلى رقم جملة FORMAT المصاحبة لجملة READ هذه (تمدنا جملة FORMAT بالمعلومات الضرورية عن نوع ومكان المتغيرات) . إن الرقم 100 هذا اختياري تماما . تأق بعد ذلك أسماء المتغيرات ويتم ذكرها بالترتيب الذي تثقب به القيم على بطاقة البيانات .

ينطبق هذا الترتيب أيضا على البنود الأربعة في جملة FORMAT المصاحبة وهي I8 و I6 و F8.0 و F12.0 . نسمي هذه البنود مواصفات الحقل (أحيانا موصف الحقل) أو كود الصيغة . يحتوى كل مدخل على حرف يدل على نوع البيانات ، إذا كان صحيحا أو حقيقيا مثلا ، ورقم يدل على عرض الحقل ، ومن ثم موقع الحقل لمنصر البيانات المناظرة . في حالتنا هذه ، تدل مواصفات الحقل الأول، I8 على أن الرقم الأول صحيح ويمكن أن يوجد في أول 8 أعمدة من البطاقة . وتدل مواصفات الحقل الثاني I6 على أن الرقم الثاني صحيح ويمكن أن يوجد في الأعمدة الستة التالية من البطاقة وتدل مواصفات الحقل الثالث F8.0 على أن الرقم الثالث حقيق ويمكن أن يوجد في الأعمدة الثمانية التالية من البطاقة وتدل المواصفات الرابعة F12.0 على أن الرقم الرابع حقيق ويمكن أن يوجد في الإثنى عشر عمودا التالية من البطاقة .

في هذا المثال ، تأخذ مواصفات حقل الرقم الصحيح الصيغة W^١ ومواصفات حقل الرقم الحقيق الصيغة F^{١٢.٠} حيث تشير W^١ إلى عرض الحقل . يفرض أن الأرقام الحقيقية تثقب بالصيغة العشرية .

٣ - ٤ الإدخال المصاغ ومواصفات حقل الإدخال

يقرأ الحاسب البيانات (المدخلات) بواسطة جملة READ تصاحبها جملة FORMAT كما تم توضيحها في الجزء الأخير ويكون الشكل العام لمثل هاتين الجملتين كما يلي :

READ(m, n) variable list
n FORMAT(format list)

بالتحديد تبدأ جملة READ بكلمة READ يتبعها عدد صحيح m يحدد رقم وحدة الإدخال وعدد صحيح n يحدد جملة FORMAT . تفصل بين m و n فصلة ويحاط الرقبن بقوسين . تحتوى قائمة المتغيرات التي تلى القوس أسماء أماكن الذاكرة التي تخزن فيها البيانات مفصولة عن بعضها بفصلات .

تبدأ جملة FORMAT المصاحبة والتي تحمل رقم الجملة n بكلمة FORMAT تتبها قائمة بمواصفات الحقل (كود الصيغة) مفصولة عن بعضها بفصلات ، وتحاط القائمة بقوسين . يجب أن تتفق مواصفات الحقل مع قائمة المتغيرات ليس فقط في الترتيب ولكن أيضا في النوع .

رغم أن جملة FORMAT تستخدم مع كل من جمل الإدخال والإخراج إلا أننا سنناقش الآن معنى مواصفات الحقل حينما تستخدم في جمل الإدخال فقط . (وسوف نناقش معناها مع جمل الإخراج في قسم ٣ - ٦) .

تحتوى مواصفات الحقل في جملة FORMAT على عرض الحقل في المدخلات بالترتيب وهي تحدد بطريقة منفردة مجموعة من الحقل المتتالية على بطاقة البيانات كما يلي :

يمثل أول حقل أول عدد W^١ من الأعمدة ، وثاني حقل عدد W^٢ من الأعمدة التالية وهكذا . ولقد كانت عروض الحقل في مثال القسم السابق 8 ، 6 ، 8 و 12 على الترتيب .

لاحظ أنه في حالة القراءة بمواصفات حقول رقمية (التي سوف نناقشها فيما بعد) تطبق القاعدتان :

١ - تصغر المسافات الخالية كأصفار .

٢ - تأخذ القيم التي لا تحمل إشارة على أنها موجبة .

إذا استخدمنا مواصفات حقول غير رقمية ، فيمكن أن نستثنى القاعدة الأولى .

حقل I :

مواصفات الحقل العامة لقراءة عدد صحيح هي

Iw

حيث يشير الحرف *I* إلى أن المتغير من النوع الصحيح ، ويشير *w* إلى رقم ثابت صحيح بدون إشارة يعطى عرض الحقل .

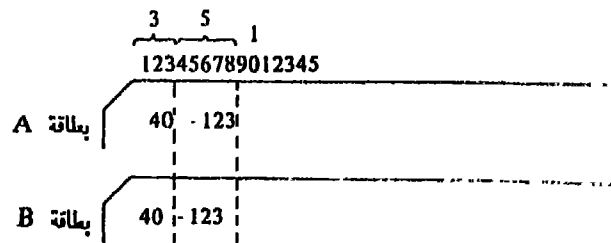
بما أن المسافات الخالية تعتبر أصفاراً في الحقول الرقمية فن الضرورى أن يتقب العدد الصحيح مضبوط من الطرف الأيمن (right-justified) في حقله بحيث تظهر آخر خانة في العدد الصحيح في آخر عمود من الحقل المعطى .

إدرس الجمل التالية :

```
READ(5, 20) M, N
20 FORMAT(I3, I5)
```

تأمر هذه الجمل الحاسب بأن يقرأ عددين صحيحين من بطاقة البيانات ويخزنها في *M* و *N* على الترتيب . مواصفات *M* هي *I3* ومواصفات *N* هي *I5* . حيث أن عروض الحقول هي 3 و 5 على الترتيب يخصص إلى *M* العدد الصحيح الموجود في الأعمدة 1 إلى 3 ويخصص إلى *N* العدد الصحيح الموجود في الأعمدة 4 إلى 8 .

افرض أن العددين الصحيحين 40 و 123 مطلوب تخزينهما في *M* و *N* على الترتيب حينئذ يجب أن تتقب بطاقة البيانات كما في البطاقة (A) من الشكل ٣ - ٢ . ومن ناحية أخرى ، لو وجب تثقيب بطاقة البيانات كما في البطاقة (B) من الشكل ٣ - ٢ لكانت جملة READ السابقة تخصص 400 إلى *M* و 1230 - إلى *N* حيث أن المسافات الخالية في الأعمدة 3 و 8 تترجم كأصفار .



شكل ٣ - ٢

حقل F :

تقرأ الثوابت الحقيقية في الشكل العشري باستخدام حقل *F* . الشكل العام هو

Fw.d

تشير *F* إلى أن البيانات من النوع الحقيقي وأن الرقم مكتوب في الشكل العشري ، حيث تشير *w* إلى عرض الحقل وتمثل بثابت صحيح

بدون إشارة ، وتمثل d عدد الخانات العشرية وهي عدد صحيح بدون إشارة أيضا . ومع ذلك ، يمكن أن تهمل d في المدخلات إذا كان العدد المنقوب في الحقل به علامة عشرية .

افترض أن المطلوب قراءة العدد 403.125 — ليخزن في A باستخدام

READ(5, 20) A
20 FORMAT(F12.4)

حيث أن عرض الحقل 12 ، فيخصص الرقم الظاهر في أول 12 عموداً من بطاقة البيانات إلى A . وعلى هذا ، يمكن أن تستخدم أيًا من البطاقات الأولى في شكل ٣ - ٣ كبطاقة للبيانات .

في الشكل العشري ، ليس من الضروري تشقيب A مضبوطة من الطرف الأيمن في الحقل الخاص بها كما هو الحال في الثوابت الرقمية الصحيحة . رغم أن المسافات الحالية في الحقول الرقمية تفسر كأصفار ، إلا أن إضافة أصفار إلى رقم في الشكل العشري من الجهتين لا يغير من قيمه ، على سبيل المثال 403.125 — و 403.12500 — لهما نفس القيمة . لذا وكما ذكر سابقاً ، فعند قراءة حقل F سيهمل الثابت d في $Fw.d$ طالما الرقم منقوب في الشكل العشري (أي يحتوي على علامة عشرية) ويقع بداخل الحقل المحدد . وسيقرأ هذا الرقم كما يظهر في بطاقات البيانات . يجب أن نلاحظ أن عدد الأرقام المعنوية التي يحتفظ بها الحاسب هي خاصية تابعة للآلة .

لا تتطلب مترجمات كثيرة أن تشقب الثوابت الحقيقية الموصفة بحقل F في المدخلات بعلامة عشرية . في هذه الحالة ، يكون الثابت الصحيح d في $Fw.d$ مؤثراً . وبالتحديد إن لم توجد علامة عشرية في الحقل المحدد ، سيكون للرقم المقروء بالضبط عدد d من الأرقام العشرية ، وتضاف علامة عشرية بين العمود رقم d والعمود رقم $(d - 1)$ بالعد من الطرف الأيمن من الحقل إلى اليسار . على سبيل المثال . افترض أننا استعملنا الزوج READ . FORMAT السابق لقراءة بطاقة 4 في الشكل ٣ - ٣ . وحيث أنه لا توجد علامة عشرية بين الأعمدة 1 و 12 ، فأثناء القراءة بمواصفات الحقل F12.4 ستعتبر علامة عشرية بين العمود رقم 4 والعمود رقم 5 من الجانب الأيمن من الحقل ، أي بين العمود 9 و 8 .

	12	1	2
	123456789012	3456789012	3456789012
(1)	-403.125		
(2)	-403.125		
(3)	-403.125		
(4)	-403125		
(5)	-403125		

شكل ٣ - ٣

الفصل الثالث : الاخلال / الاخراج العددي

٧

بالتالى ، فيكون العدد المخصص لـ A هو 40.3125 - وبالمثل ، لو أننا استعملنا نفس الزوج READ-FORMAT لفرادة بطاقة 5 من الشكل ٢ - ١ لكان العدد المخصص لـ A هو 4031.25 - .

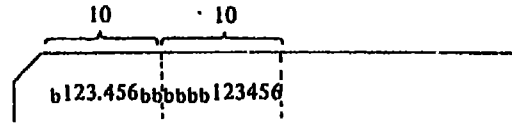
تهمل السمة السابقة الحاجة لتثقيب علامة عشرية في الثوابت الحقيقية عند تحضير مجموعة بطاقات البيانات (واضح أن هذا سؤا الوقت للتجهيزات التجارية) ومع ذلك عند استخدام هذه السمة فان مكان الرقم المثقب يؤثر على الرقم المخصص . وبالتالى ننصح بتطلي البرامج المبتدئين بالتنازل عن هذه السمة ، أى الإلتزام بتثقيب العلامة العشرية عند استخدام حقول F- .

مثال ٣ - ٢

افرض أن زوج READ-FORMAT التالى :

```
READ(5, 21) A, B
21 FORMAT(F10.2, F10.2)
```

لأستخدام لقراءة بطاقة مثقبة كالتالى :



حيث أن الرقم في الأعمدة من 1 إلى 10 يحتوى على علامة عشرية ، فانه يخصص لـ A كما يظهر ، ومن ثم ستحتوى A على 123.456 وأن الرقم الثانى في الأعمدة 11 إلى 20 لا يحتوى على علامة عشرية ، ومن ثم سيقراً ويخصص لـ B 123.56

حقل E :

تذكر أولاً ، أن الثابت الحقيقي والشكل الأسى (قسم ٢ - ٢) يتكون من جزئين الجزء الأول ثابت حقيقى في الشكل العشرى ويتبع هذا الجزء الأسى (الجزء الثانى) ويبدأ بالحرف E وقيمته ثابت صحيح له إشارة أو بدون إشارة ولكن بعد أقصى عددان .

تقرأ الثوابت الحقيقية في الشكل الأسى باستخدام حقل E . الشكل العام لمواصفات الحقل هو :

Ew.d

يدل الحرف E على أن البيانات حقيقية ، ويكتب الثابت في الشكل الإسى ، تدل w على عدد صحيح بدون إشارة على عرض الحقل . وتدل d ، على عدد صحيح بدون إشارة على عدد الأرقام العشرية . كما في حقل E- تهمل d في المدخلات إذا تم تثقيب الرقم بعلامة عشرية .

افرض أن المطلوب قراءة الرقم 02 : E4.03125E-02 وتخزينه في A باستخدام

```
READ(5, 30) A
30 FORMAT(E18.5)
```

لاحظ أن عرض الحقل هو 18 وعلى هذا فسوف يخصص الرقم الظاهر في أول 18 عموداً من بطاقة البيانات لـ A . وبذلك يمكن استخدام أى بطاقة من أول خمس بطاقات في الشكل ٣ - ٤ كبطاقة بيانات لتعطي A القيمة $4.03125 \cdot 10^3$ - . إذا استخدمت البطاقة السادسة سوف يتم تخصيص الرقم 20 لـ A حيث تقسم المسافة الخالية في العمود 18 كسفر وبالتالى من الضروري أن يثقب الجزء الأسى مضبوطاً من جهة اليمين في حقل E- .

إذا كان ثابت حقيقى له إس صفر (أى 0.0 + E) حينئذ يمكن عدم تثقيب جزء الأس . بمعنى آخر إذا تم تثقيب الرقم بغير الجزء الأسى ، فإدلى مواصفات الحقل Ew.d بنفس طريقة Fw.d وبالتالى يمكن أن تستخدم البطاقة السابعة من الشكل ٣ - ٤ لتخصص

4.03125 E + 02 إلى A (في ضوء هذه المرونة ، يفضل بعض نمطى البرامج استخدام حقل E في المدخلات بغض النظر إذا كانت بطاقات البيانات مشتقة بأس أم لا)

	18																	
	1									2								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
(1)	-4.03125E+02																	
(2)	-.403125									E+ 3								
(3)	-40.3125E+1																	
(4)	-4.03125									E2								
(5)	-4031.25									E-1								
(6)	-4.03125									E2								
(7)	-403.125																	
(8)	-403125E+2																	
(9)	-403125									E2								
(10)	-403125									E1								

شكل ٣ - ٤

كما في حقل F- فاستخدام العلامة العشرية في المدخلات اختياري عند استخدام حقل E . إن لم يكن هناك علامة عشرية في الحقل المحدد (على يسار E) سنفترض علامة عشرية بين العمود d و d+1 مبتدأ بالعدد من أول عمود على يسار الحرف E متجهاً إلى اليسار . على سبيل المثال ، افترض أننا استخدمنا الزوج السابق READ-FORMAT لقراءة بطاقة (8) من الشكل ٣ - ٤ . ولعدم وجود علامة عشرية في الأعمدة 1 إلى 18 فسوف يتسبب العدد الصحيح 5 في المواضع E18.5 في إضافة علامة عشرية بين العمود الخامس والسادس من يسار الحرف E أى بين العمود 10 و 11 . وبالتالي فلا يزال الرقم المخصص له A هو 4.03125E + 2 . من ناحية أخرى ، لو استخدمنا نفس الزوج READ-FORMAT لقراءة البطاقة (9) من الشكل ٣ - ٤ فسيخصص

40.3125E2 - ل A حيث تضاف العلامة العشرية بين العمود 11 و 12 (سيكون الحرف E هنا في العمود 17) وبذلك فسوف يتخصص الرقم 40.3125E1 - ل A إذا استخدمنا بطاقة (10) .

حقل X

يمكن أن نميز بين عدة ثوابت على نفس بطاقة البيانات وذلك بفصلهم بواسطة مسافة خالية أو أكثر . ويمكن تخطي حد المسافات باستخدام حقل -X . وبصورة عامة ، مواصفات حقل -X تأخذ الشكل :

n هي عدد صحيح بدون إشارة وتشير إلى عرض الحقل ، ويعني الحرف X أنه يجب تخطي الحقل المناظر المتكون من عدد n من الأعمدة بغض النظر عما إذا كان الحقل يحتوي على معلومات أم لا .

مثال ٣ - ٣

افترض أن برنامج به الجمل

READS, 75) ID, AMOUNT, BALANS
75 FORMAT(I7, 3X, F7.2, 3X, E12.2)

وافترض أنه قد تم قراءة بطاقة البيانات في الشكل ٣ - ٥ . لاحظ أن عروض الحقول هي 7 ، 3 ، 7 ، 3 ، 12 على الترتيب . وعلى هذا ، سيخصص إلى ID العدد الصحيح في أول حقل رقمي (الأعمدة 1 إلى 7) وسوف تخطى الأعمدة 8 إلى 10 . ثم يخصص الرقم الحقيقي في ثاني حقل رقمي إلى AMOUNT (الأعمدة 11 إلى 17) وسوف تخطى الأعمدة 18 إلى 20 . بعد ذلك يخصص الرقم الحقيقي في ثالث حقل رقمي إلى BALANS (الأعمدة 21 إلى 32) وعلى ذلك :

D = 12345, AMOUNT = 250.25, BALANS = 402.5 = 40.25E+1

7	3	7	3	12
1		2		3
12345678901234567890123456789012345				
12345678		250.2567		40.25E+1

شكل ٣ - ٥

٣ - ٥ جمل WRITE المصاغة وتحكم العسرية

يأمر الحاسب بإخراج البيانات بواسطة جملة WRITE مصحوبة بجملة FORMAT وتأخذ الشكل التالي :

WRITE(m, n) variable list
n FORMAT(format list)

وبالتحديد فإن جملة WRITE تبدأ بكلمة WRITE . يتبعها عدد صحيح m يحدد رقم وحدة الإخراج ورقم جملة n تحدد جملة FORMAT المصاحبة ، تفصل بين m و n فصلة ويماطا بقوسين وعلى ذلك المتغيرات المطلوب طباعة قيمها مفصولة بفواصل . ستفترض أن آلة الطباعة هي التي تستخدم لطباعة مخرجاتنا . لذا فرقم وحدة الإخراج في كل أمثلتنا سيكون $m = 6$ (انظر جدول ٣ - ١) .

يجب أن تكون جملة FORMAT المصاحبة معنونة برقم الجملة n وستأخذ نفس الشكل كما في المدخلات . أي أنها تبدأ بكلمة FORMAT متبوعة بقائمة من مواصفات الحقول منفصلة عن بعضها بواسطة فاصلات وتماط بأقواس . وسنشرح معاني مواصفات الحقول في الإخراج في القسم التالي .

سيولد كل زوج من WRITE-FORMAT سيل من حروف الإخراج (سجل) . (المسافة الخالية هي حرف وسيرمز إليها أحياناً بـ « b » ، أي b كدليل) . يتحكم أول حرف من سجل الإخراج في حركة عربة آلة الطباعة تبعاً للتعليمات في جدول ٣ - ٢ ولكنه لا يطبع . بالتحديد يختبر الحاسب أول حرف من سجل الإخراج ثم يؤدي أمر التحكم في العربة ، ثم يطبع بعد ذلك بقية الحروف في سجل الإخراج .

جدول ٣ - ٢

مدخل الصياغة	تعليمات التحكم	الحروف الأولى
'IX or'	تقدم سطر واحد (في العادة مسافة سطر واحد)	(مسافة) b
'0'	تقدم سطرين (يتخطى ، سطر الكتابة على مسافتين)	0 (صفر)
'1'	تقدم إلى أول الصفحة التالية	١
'+'	لا تتقدم	+

على سبيل المثال ، سجل الإخراج .

bbbb12345bb67.89

يأمر الحاسب بالتقدم سطر واحد ثم يطبع بعد ذلك 4 مسافات خالية ثم 12345 ، ثم مسافتين خاليتين و 67.89 ، وسجل الإخراج

0bbbb12345bb67.89

يأمر الحاسب بالتقدم سطرين (يتخطى سطر) ثم يطبع 4 مسافات خالية ثم 12345 ، ثم مسافتين خاليتين ثم 67.89 ، وسجل الإخراج

1bbbb12345bb67.89

ويأمر الحاسب بالتقدم إلى صفحة جديدة ثم يطبع 4 مسافات خالية ثم 12345 ثم مسافتين خاليتين ثم 67.89 .

تعد جملة FORMAT في الزوج WRITE-FORMAT الحاسب بأنواع بيانات الإخراج ومكان ظهورها في سجل الإخراج . بالتحديد يخصص لكل قيمة في قائمة التنزيات في جملة WRITE مجموعة من المسافات (تسمى حقلاً) . ذلك عن طريق مدخل مناظر في جملة FORMAT . توضع دائماً القيمة العددية للمتغير في الحقل الخاص بها .

مضبوطة من الطرف الأيمن (right-justified) .

أي ينتهي الرقم في آخر مسافة خالية من الحقل . وسوف نوضح ذلك بمثال رغم أننا سنناقش مداخل الصيغة بالتفصيل في القسم التالي .

مثال ٣ - ٤

اوترس أساساً دنا طباعة القيم ل . K ، L بالترتيب $J = 128$ ، $K = 179$ ، $L =$ يمكن إنجاز ذلك بزواج الحقل أدنى .

```
WRITE(6, 20) J, K, L
20 FORMAT(18, I6, 18)
```

يخبر الرقم 6 في جملة WRITE الحاسب أن وحدة الإخراج هي آلة الطباعة ، ويدل الرقم 20 على أن القيم ستطبع سبقتاً بجملة FORMAT التي تحمل الرقم 20 (يختار الرقم 20 عشوائياً) .

تحدد أول ثلاثة حقول في سجل الإخراج بواسطة زوج WRITE-FORMAT الذي يحتوي الحروف التالية :

bbbb128 (18) تتولد القيمة الصحيحة ل J مضطمة من الطرف الأيمن من أول حقل بعرض 8 : أي أن قيمة J مرودة من اليسار بخمسة حروف خالية .

bbb72 (16) تتولد القيمة الصحيحة ل K مضطمة من الطرف الأيمن من الحقل التالى بعرض 6

bbb-179 (18) تتولد القيمة الصحيحة ل L مضطمة من الطرف الأيمن من الحقل التالى بعرض 8 .

وبالتالى ، يحتوى سجل الإخراج على الحروف التالية :

سجل الإخراج ، 79 ، bbbb128bbb72bbb

وعلاوة على ذلك ، يخبر سجل الإخراج الحاسب بالتقدم سطرأ واحداً ، ثم يطبع بعد ذلك 4 مسافات ثم 128 و 4 مسافات ، ثم 72 ، و 4 مسافات ثم 179 . وتظهر المخرجات المطبوعة في الشكل ٣ - ٦ . صفحات الطباعة لها عروض مختلفة ، وصفحة الطباعة في الشكل ٣ - ٦ تقترن أن هناك 132 عموداً للطباعة .

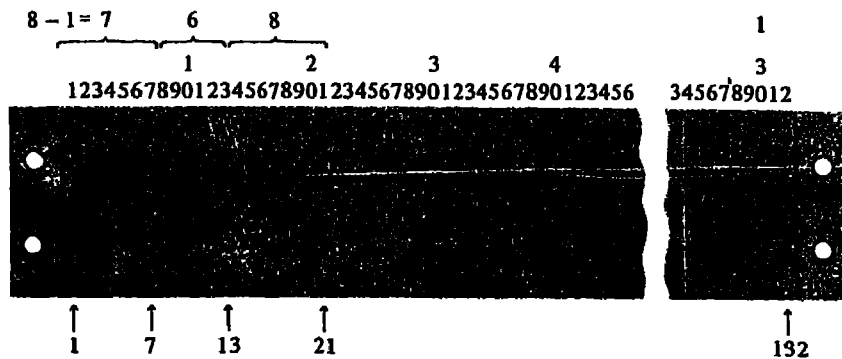
تلخيصاً لما سبق يؤدي الحاسب الخطوات التالية في أمر WRITE-FORMAT :

١ - يولد سجل الإخراج المنشأ بزواج WRITE-FORMAT

٢ - يختبر أول حرف وينفذ أمر التحكم في العربة .

٣ - يطبع بقية الحروف من سجل الإخراج

نلاحظ أنه بعد إتمام أمر WRITE تتوقف آلة الطباعة دائماً على السطر الذى تمت طباعته . ومن ثم ، فأول حرف مولد بواسطة جملة إخراج تالية سيستخدم مرة أخرى للتحكم في العربة .



شكل ٣ - ٦

ملحوظة : من العادات الجيدة في البرمجة أن يعطى الفرد عادة إشارة صريحة للتحكم في العربة . يتم هذا باختيار أحد المدخلات في جدول ٣ - ٢ كمدخل أول في جملة FORMAT . سنناقش ونشرح هذا في القسم التالي (بهذه العادة يمكن تجنب الأخطاء .
الموضحة في المسائل ١٢ - ٢ (ب) ، ٣٢ - ٢ (و) ، و ٣٢ - ٢ (و)) . لاحظ أن المثال السابق لا يحتوى على إشارة صريحة للتحكم في العربة ، ولكن إشارة التحكم في العربة كانت متضمنة في مواصفات الحقل 18 التي أعطت حقلاً أكبر من القيمة l .

٣ - ٦ مواصفات حقل الإخراج

كما نوقش من قبل ، فإن جملة FORMAT المصاحبة لجملة WRITE تحدد أنواع البيانات وأماكن المعلومات التي ستطبع . في هذا القسم سوف نغطي نفس أنواع مواصفات الحقل التي غطيت في الإدخال .

wX , Iw , $Fw.d$, $Ew.d$

ويؤم التأكيد بأن هذه المواصفات في الإخراج قد تختلف عنها في الإدخال . والسبب الأول في ذلك أن كل الثوابت الرقمية تطبع مضطبة من الطرف الأيمن في الحقول الخاصة بها ومضاف إليها على اليسار مسافات خالية . كذلك ، تلب d في $Fw.d$ و $Ew.d$ دوراً هاماً في l ، في حين أن d قد تلتفى أحياناً في المدخلات .

حقل X

يمكن أن نحصل على مسافات خالية للفصل بين عدة ثوابت على نفس السطر باستعمال حقل X . الشكل العام لمواصفات الحقل هو :

wX

حيث تدل w على عرض الحقل . وتأثير wX هو توليد عدد w من المسافات الخالية في سجل الإخراج .

وعلى وجه الخصوص ، إذا ظهرت IX كأول مدخل في جملة FORMAT حينئذ تتولد مسافة كأول حرف في سجل الإخراج . تستخدم هذه المسافة للتحكم في العربة فتجعلها تتقدم سطرًا واحداً . بمعنى آخر ، إذا كانت جملة FORMAT في الشكل

$n \text{ FORMAT}(IX, \dots)$

فإنها تؤدي دائماً إلى مسافة واحدة بين السطور ، بالمثل .

$n \text{ FORMAT}(10X, \dots)$

تتبع دائماً مسافة واحدة بين السطور مع الزحزحة تسعة مسافات للداخل .

حقل I

تطبع الثوابت الصحيحة باستعمال حقل I . الشكل العام لمواصفات الحقل هو :

Iw

حيث تدل I على النوع الصحيح و w عرض الحقل .

على سبيل المثال ، افترض أن M تحتوي 250 و N تحتوي 46 وتم تنفيذ

WRITE(6, 10) M, N
10 FORMAT(1X, I6, 3X, I4)

يولد المدخل IX مسافة واحدة ، يولد المدخل 16 الذي يباشر M 250 bbb أى أن قيمة M تتولد مضبوطة من الطرف الأيمن من الحقل التالى ذى العرض 6 . يولد المدخل 3X ثلاث مسافات bbb يولد المدخل 14 الذي يباشر N 46 bb أى أن قيمة N تتولد مضبوطة من الطرف الأيمن . احس أنت ذى العرض 4 وعلى هذا ، يتولد سجل الإخراج التالى .

سجل الإخراج : `bbb250bbb46`

وحيث أن أول حرف من سجل الإخراج هو مسافة خالية وذلك يعنى تباعداً مفرداً للسطور ، فستتدم آلة الطباعة سطرأ واحداً وبعيد بقية الحروف :

أحرف المطبوعة : `bbb250bbb46`

إذا لم يتوافر عرضاً كافياً ، أى إذا احتوت قيمة رقمية على حروف أكثر من الحروف الموصوفة في عرض الحقل ، ستر بعض المترجمات القيمة من اليسار (أو اليمين) وتطبع بقية الحروف . في المثال السابق ، لو احس N على 54321 إذن ستطبع :

`bbb250bbb4321`

ولا تعلى أى رسالة لنشير إلى ذلك . لحس الحظ ، فإن معظم المترجمات الميئة بطريقة مناسبة وستشير إلى نقص عرض الحقل تطبع نجوم في الحقل الميئ . باستعمال المثال السابق مرة أخرى مع تحديد $M = 54321$ فسوف يطبع

`bbb250bbb****`

يجب أن يراعى مخطوطو البرامج أن يكون عرض الحقل كبيراً بدرجة كافية لتلائم كل قيم المتغيرات الممكنة .

حقل F- :

يمكن أن تطبع الثوابت الحقيقية في الشكل العشري باستعمال حقل F- . الشكل العام لمواصفات الحقل هو :

Fw.d

حيث تشير F إلى نوع البيانات و w إلى عرض الحقل . يشير العدد الصحيح d إلى عدد الأرقام العشرية ويتم هذا دائماً بالتقريب . بمعنى آخر ، تخبر Fw.d الحاسب بأن يقرب الرقم الحقيقي المناظر إلى d رقماً عشرياً ، وأن يطبع الرقم مضبوطة من الطرف الأيمن في الحقل الخاص به ذى العرض w .

مثال ٣ - ٥

افرض $ID = 125$ و $AMOUNT = 450.2462$ وتم تنفيذ

```
WRITE(6, 27) ID, AMOUNT
27 FORMAT(I8, 3X, F10.2)
```

لاحظ أن مواصفات الحقول تحتوي على عرض الحقل 8 ، 3 ، 10 على الترتيب . بحس مواصفات الحقل I8 والتي تناظر ID الحاسب بأن يولد القيمة ID مضبوطة من الطرف الأيمن في أول حقل ذى العرض 8 من سجل الإخراج ، لذا تتولد `bbbbbb125` تولد 3X ثلاث مسافات خالية : `bbb` . تخبر F10.2 الحاسب أن يقرب قيمة A إلى رقين عشريين أى إلى 450.25 وأن يولد هذه القيمة مضبوطة من الطرف الأيمن في الحقل التالى ذى العرض 10 لذا تتولد `bbb450.25` وبذلك يصبح سجل الإخراج الذى تم توليده :

سجل الإخراج : `bbb125bbb450.25`

حيث يستخدم أول حرف للتحكم في العربة ، سيظهر الحرج على صفحة الطباعة كما في الشكل ٣ - ٨ .
وأنا لتؤكد أن الرقم 450.25 يشغل ست مسافات في الحرج حيث أن العلامة العشرية تشغل مسافة طباعة واحدة .



شكل ٣ - ٧

كما في حقل I- إن لم يتح عرض كاف ، أى ، إذا كانت القيمة العددية في حقل F- تحتوي حروفاً أكثر من عرض الحقل ، قد يبرز الرقم من اليسار (أو اليمين) أو قد تطبع نجمة . وعلى هذا ، يجب أن نتوخى الحرس عند استعمال حقل F- وأن يكون عرض الحقل كبيراً بدرجة كافية ليلائم الرقم بالتحديد ، أن كان العدد المطلوب من الأرقام العشرية هو d و n هي عدد خانات الجزء الصحيح ، فنل الأقل يجب أن يكون $w \geq n + d + 2$. تسمح 2 بمسافة للإشارة ومسافة للعلامة العشرية . على سبيل المثال إذا كانت $|X| < 1000$ ونحن نحتاج إلى رقمين عشريين فقط ، إذن تكون F7.2 كافية : $w = 3 + 2 + 2$ حيث تتاح ثلاث مسافات للجزء الصحيح من X .

يجب أن نتوخى الحرس حيث عدد الأرقام العشرية كبيراً بدرجة كافية . على سبيل المثال ، افرض أن A تحتوي على 0.246E-05 ومواصفات الحقل هي F12.4 (تخزن الأرقام الحقيقية في الحاسب بالشكل العلى ، أنظر ملحق أ) . حينئذ يطبع .

0.0000

فقط حيث أن قيمة A ، وهي 0.00000246 في الشكل العشرى ستكون 0.0000 عندما تقرب إلى أربعة أرقام عشرية .

حقل E

يمكن أن نطبع الثوابت الحقيقية في الشكل الأسى باستعمال حقل E- . الشكل العام لمواصفات الحقل هو

 $Ew.d$

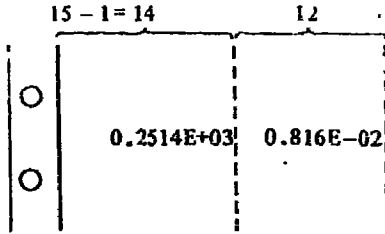
حيث تشير E إلى نوع البيانات وتشير w إلى عرض الحقل . يشير العدد الصحيح d إلى عدد الأرقام المعنوية ويتم هذا بالتقريب . بالإضافة إلى ذلك ، تطبع الثوابت الحقيقية في الحرج بالشكل الأسى المعيارى ، أى كثوابت بين 0.1 إلى 1.0 (- 1.0 إلى بصحة الأس المناسب الذى يحتوى دائماً على أربعة حروف $E \pm XX$ بمعنى آخر تختبر Ew.d الحاسب أن يقرب الرقم الحقيقى المناظر إلى العدد d من الأرقام معنوية وأن يطبع العدد في الشكل الأسى المعيارى مضبوطاً من الطرف الأيمن في الحقل الخاص به ذى العرض w . (ونلاحظ أن في الشكل الأسى المعيارى ، عدد الأرقام المعنوية مساوى أيضاً لعدد الأرقام العشرية)

مثال ٣ - ٦

افرض $A = 251.381$ و $B = 0.0081626$ وافرض أنه تم تنفيذ

WRITE(6, 22) A, B
22 FORMAT(E15.4, E12.3)

وحيث أن الأرقام الحقيقية (في الشكل E أو F) تخزن داخلياً في الشكل الأسّي المعياري (أنظر ملحق أ) . تحتوي A على $0.251381 E+03$ وتحتوي B على $0.81626 E-02$. يخبر المدخل $E15.4$ الحاسب أن يقرب قيمة A إلى أربعة أرقام منوية أي $0.2514E+03$ ويضعها مضبوطة من الطرف الأيمن في الـ 15 مسافة الأولى من سجل الإخراج . ويخبر المدخل $E12.3$ الحاسب أن يقرب قيمة B إلى ثلاثة أرقام منوية أي $0.816E-02$ ، ويضعها مضبوطة من الطرف الأيمن في 12 مسافة التالية من سجل الإخراج . وبالتالي فإنها تظهر كما هو موضح في الشكل ٣ - ٨ .

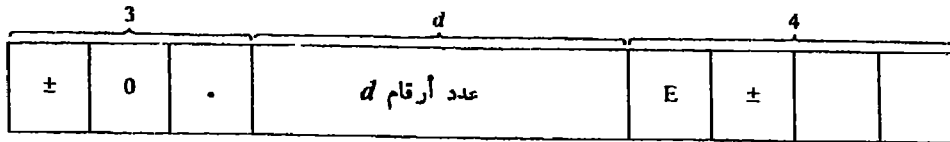


شكل ٣ - ٨

لإتاحة عرض كاف لطبع الأرقام العشرية في شكل E- فإننا نحسب w كما يلي : إذا كان المطلوب عدد d من الخانات المنوية يجب أن يحقق عرض الحقل w الشرط التالي :

$$w \geq d + 7$$

ويمكن الحصول عليه كالتالي : أربع مسافات مطلوبة للأس $E \pm XX$ ، مسافة واحدة للإشارة ومسافة واحدة للصفر الموجود على اليسار ، مسافة واحدة للعلامة العشرية . وبين شكل ٣ - ٩ أيضاً طريقة الحساب .



شكل ٣ - ٩

نلخص قسمي الإدخال والإخراج بالتعليقات التالية :

- ١ - يمكن أن تحتوي جملة FORMAT على كثير من مواصفات الحقول . ويجب أن تكون الحقول متلاحقة ، أي تلي بعضها الآخر .
- ٢ - لقراءة قيم أو طباعتها ، يجب أن يكون هناك تناظر في ترتيب وضع المتغيرات مع ترتيب مواصفات الحقول الرقمية (أو الحرفية) في جملة FORMAT المناظرة ويجب أن تتفق أيضاً في النوع .
- ٣ - يجب أن تصاحب كل جملة READ أو WRITE جملة FORMAT ومع ذلك يمكن لجملة FORMAT واحدة أن تصاحب (يشار إليها من) عدة جمل I/O . وعلاوة على ذلك ، فإن جمل FORMAT جمل غير منفذة .

٣ - ٧ الحقل الحرفي

افترض أننا نرغب في طباعة عنوان POPULATION OF USA . فلإنجاز ذلك نهيئ الرسالة بفصلات عليا داخل جملة FORMAT كالتالي :

```
WRITE(6, 30)
30 FORMAT(1X, 'POPULATION OF USA')
```

٥ - البرمجة بلغة الفورتران

عند تنفيذ جملة WRITE السابقة ، سيتم توليد كل الحروف بداخل الفاصلات العليا . ويحتوى سجل الإخراج :

POPULATION OF USA : سجل الإخراج

حيث تولد المسافة الأول بسبب 1X . آخذين في الاعتبار التحكم في العربية ، بذلك تطبع الرسالة POPULATION OF USA

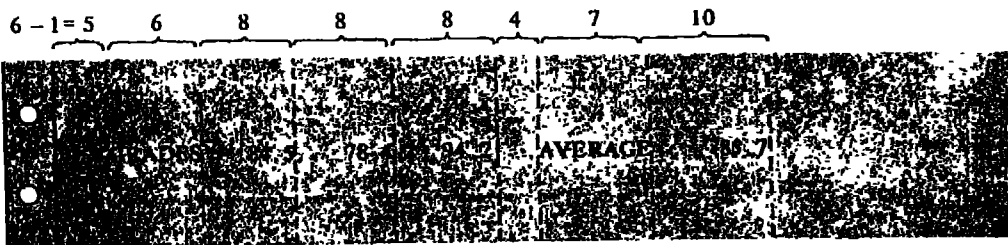
يمكن أن تطبع رسالة حرفية مع قيم رقمية . افرض $G1 = 84.5$ ، $G2 = 78.4$ ، $G3 = 94.2$ ، $AVE = 85.7$. فسوف تولد التعليقات :

WRITE(6, 80) G1, G2, G3, AVE
80 FORMAT(6X, 'GRADES', 3F8.1, 4X, 'AVERAGE', F10.1)

سجل الإخراج التالي :

GRADES 84.5 78.4 94.2 AVERAGE 85.7 : سجل الإخراج

سوف يظهر المخرج على ورقة الطباعة كما في الشكل ٣ - ١٠ . لاحظ أن المدخل 6X سيتسبب في طباعة 5 - 1 مسافات خالية فقط ، حيث تستخدم أول مسافة للتحكم في العربية . لاحظ أيضاً أن الرسائل GRADES و AVERAGE تشغل حقولاً ذات عروض 6 و 7 على الترتيب ، وهي عدد الحروف في الرسالة .



شكل ٣ - ١٠

(لاحظ أن 3F8.1 هي اختصار F8.1, F8.1, F8.1 سوف نذكر ذلك في قسم ٣ - ٩) .

أحد الاستخدامات الأساسية للمقل الحرفي هو التحكم في العربية . افرض أننا أردنا طباعة الرسالة POPULATION OF USA في بداية صفحة جديدة . يمكن إنجاز ذلك بواسطة الجمل :

WRITE(6, 20)
20 FORMAT('1', 'POPULATION OF USA')

ويكون أول مدخل للصيغة 1 هو أمر التحكم في العربية ليذهب إلى صفحة جديدة . بالمثل . يمكن أن تستخدم '0' كأول مدخل للصيغة إذا أردنا أن يتخطى الحاسب سطرًا قبل طباعة POPULATION OF USA أو سوف نستخدم '+' كأول مدخل للصيغة . إذا أردنا أن يطبع الحاسب على السطر الجالي . توجد قائمة بهذه المداخل في أماكنها المناسبة في العمود الثالث من جدول ٣ - ٢ الذي يحتوي على تعليقات التحكم في العربية .

٣ - ٨ السجلات ، السجلات المتعددة ، الشرطة المائلة (/)

السجل هو مجموعة البيانات المناظرة للمتغيرات في قائمة (I/O) . والحد الأقصى لطول السجل محدود بوحدة (I/O) التي ذكرناها في هذا الشرح وهي : 80 حرف البطاقات ، 72 حرف لالة الكاتبة المركزية ، 132 حرف لوحدة الطباعة . بالطبع ، فإن الطول الحقيقي يستمد من أمر (I/O) المعطى في جملة FORMAT .

حتى الآن فإن كل أمر من أوامر (I/O) نقل سجل واحد من المعلومات . يقرأ زوج الجمل `READ-FORMAT` من بطاقة إدخال واحدة ويطبع كل زوج جمل `WRITE-FORMAT` على سطر مخرج واحد . إلا أنه ، يمكن أن تستخدم جملة `FORMAT` واحدة لتعريف سجلات متعددة . وسنناقش فيما يلي طريقتان من هذه الطرق :

الحالة (أ)

تذكر أن تنفيذ جملة `READ/WRITE` لا تتم إلا إذا كانت كل المتغيرات المذكورة قد أعطيت قيم أو طبعت . افترض أنه تم تنفيذ

```
READ(5, 20) A, B, C, D
20 FORMAT(F8.2, 2X, F15.7)
```

لاحظ أن عدد المتغيرات (4) وذلك أكثر من المواصفات الرقمية (2) . وبذلك فإننا نقرأ المتغيرين `A` و `B` من بطاقة 1 أولاً من الأعمدة 1 إلى 8 و 11 إلى 25 بالمواصفات `F8.2` و `F15.7` على الترتيب . وحيث أن هناك متغيرات أخرى يجب قراءتها ، فسوف تستخدم نفس جملة `FORMAT` مرة ثانية ، ولكن مبتدأة بسجل جديد ، أي بطاقة أخرى . ومن ثم ، نقرأ قيم `C` و `D` من بطاقة 2 من الأعمدة 1 إلى 8 و 11 إلى 25 على الترتيب (أنظر أيضاً قسم ١٠ - ٦) .

بالمثل ، سوف تنتج الجملتين

```
WRITE(6, 30) A, B, C, D
30 FORMAT(5X, F8.2, 2X, E14.7)
```

سجلين للمخرجات .

```
hhhh + XXXX.XXhh + 0.XXXXXXXXXE + XX
hhhh + XXXX.XXhh + 0.XXXXXXXXXE + XX
```

ومن ثم ، ستطبع `A` و `B` على سطر واحد تبعاً لـ `F8.2` و `E14.7` على الترتيب مع مراعاة ترك 4 مسافات إلى الداخل . وستطبع `C` و `D` على السطر التالي بنفس الطريقة .

ملحوظة : في الحالة العكسية حيث توجد هناك مواصفات حقول أكثر من المتغيرات . فيتم تنفيذ جملة `READ/WRITE` بمجرد أن تقرأ / تطبع كل المتغيرات وبالتالي تهمل بقية المواصفات الزائدة . على سبيل المثال إذا غيرنا جملة `READ` السابقة إلى :

```
READ(5, 20) A, B, C, D, E
```

وبذلك تقرأ ثلاث بطاقات وسيخصص الرقم الموجود في الأعمدة 1 إلى 8 من البطاقة الثالثة إلى `E` باستعمال `F8.2` .

الحالة (ب) :

يمكن أيضاً أن نستخدم الشروط المائلة (/) لتعريف السجلات المتعددة . تستخدم الشروط المائلة (/) في جملة `FORMAT` للإشارة إلى إنهاء السجل ، وعلى هذا ، تقرأ الجملتان التاليتان سجلين :

```
READ(5, 40) A, B, C, D
40 FORMAT(F10.2, F10.2/E15.7, E15.7)
```

أي تقرأ `A` و `B` من بطاقة 1 باستعمال `F10.2` و تقرأ `C` و `D` من بطاقة 2 باستعمال `E15.7` . بالمثل فإن الجملتين :

```
WRITE(6, 50) N, AVE
50 FORMAT('1', 'NUMBER OF STUDENTS =', I3/1X, 'EXAM AVERAGE =', F7.2)
```

تعليمان سجلين وسيطع في نهاية صفحة جديدة السطرين التاليين :

```
NUMBER OF STUDENTS = XXX
EXAM AVERAGE = XXX.XX
```

ونؤكد أنه يجب أن نذكر حرف التحكم في العربة لكل سجل خرج جديد . وبالتحديد فان 1X التي تلي الشرطة المائلة في جملة FORMAT السابقة هي أمر التحكم في العربة للسجل الجديد .

يمكن أيضاً أن تظهر عدة شرطات مائلة متتالية - إلا أن ، الشرطات المائلة الظاهرة في آخر جملة FORMAT تعطي نتيجة مختلفة اختلافاً بسيطاً عن التي تظهر في سجل الإدخال وبالتحديد :

١ - في الإخراج . عند ظهور عدد n من الشرطات المائلة المتتالية ، فسوف تطبع عدد $n - 1$ من الأسطر الحالية وعلى سبيل المثال كلا من

```
WRITE(6, 20) J, K           و           WRITE(6, 10) J, K
20 FORMAT(1X, I10//)       10 FORMAT(1X, I10//1X, I10)
```

سينتج سطران خاليان بين J و K .

٢ - في الإدخال عند ظهور عدد n من الشرطات المائلة المتتالية في الوسط فسوف يتخطى عدد $n - 1$ من البطاقات ، ولكن ، عند ظهور عدد n من الشرطات المائلة المتتالية في النهاية فسوف يتخطى عدد n من البطاقات وعلى سبيل المثال فسوف يتسبب :

```
READ(6, 30) J, K
30 FORMAT(I10//I10)
```

في قراءة J من البطاقة الأولى وقراءة K من البطاقة الرابعة أي أنه سوف يتخطى بطاقتين . بينما سوف يتسبب :

```
READ(6, 40) J, K
40 FORMAT(I10//)
```

في قراءة J من البطاقة الأولى وقراءة K من البطاقة الخامسة أي أنه سوف يتخطى ثلاث كروت (أنظر المسائل ٣ - ٣٩ و ٤٠ - أيضاً) .

٣ - ٩ معامل التكرار

إفرض أن المطلوب قراءة / كتابة أربعة أرقام A و B و C و D باستعمال نفس المواصفات F10.2 فيجب أن تكون جملة FORMAT هي

```
FORMAT(F10.2, F10.2, F10.2, F10.2)
```

ولتبسيط ذلك يمكن أن نستخدم معامل التكرار 4 :

```
FORMAT(4F10.2)
```

وبذلك تكون جملةنا FORMAT لها نفس المعنى تماماً . بالمثل الجملتين التاليتين :

```
FORMAT(2I4, 3E15.7)
```


و

FORMAT(I4, I4, E15.7, E15.7, E15.7)

كما نفس المعنى .

يمكن أيضاً أن نكرر مجموعة من مواصفات الحقول وعلى سبيل المثال :

FORMAT(5X, 2(I5, 3X, F8.2, 2X))

تعني .

FORMAT(5X, I5, 3X, F8.2, 2X, I5, 3X, F8.2, 2X)

والجمللة :

FORMAT(3(3X, I4), 2(5X, E9.2))

هي نفس الشيء مثل .

FORMAT(3X, I4, 3X, I4, 3X, I4, 5X, E9.2, 5X, E9.2)

وبهذه الاختصارات تكون جملة FORMAT أكثر إيجازاً وأسهل كتابة .

٣ - ١٠ برنامج بسيط كامل

لقد غطينا مادة كافية تمكنا من كتابة بعض البرامج الكاملة البسيطة . أحد هذه البرامج مكتوب هنا واليغفر الآخر مطبوع في قسم المسائل المحلولة .

افترض أن أطوال أضلاع مثلث T هي a و b و c والمحيط P للمثلث هو مجموع هذه الأطوال :

$$P = a + b + c$$

تعطى المساحة T بالمعادلة :

$$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = P/2 = (a + b + c)/2.$$

حيث

افترض أننا نريد أن نحسب محيط ومساحة مثلث أطوال أضلاعه 38.6 و 42.4 و 56.1 سوف نكتب أولاً هذه الأرقام على بطاقة بيانات باستعمال ، حقول بمرص 10 مثلاً ، كما في شكل ٣ - ١١ .

1	2	3
12345678901234567890123456789012345		
38.6	42.4	56.1

شكل ٣ - ١١

برنامج الفورتران الذى يجرى الحسابات السابقة هو :

```

C
C PROGRAM CALCULATING PERIMETER AND AREA OF A TRIANGLE
C
READ(5, 11) A, B, C
11 FORMAT(3F10.1)
P = A + B + C
S = P/2.0
AREA = SQRT(S*(S - A)*(S - B)*(S - C))
WRITE(6, 12) A, B, C, P, AREA
12 FORMAT(3(F8.1, 4X), F10.1, 4X, F12.2)
STOP
END

```

شرح وتنفيذ البرنامج

الأسطر ١ - ٣ هذه الأسطر تعليق حيث أن C مثقبة في أول عمود .
الأسطر ٤ - ٥ تجبر هذه الأسطر الحاسب أن يقرأ قيم A و B و C من بطاقة البيانات باستعمال حقول كل منها بعرض 10 . ومن ثم ، تخصص القيم الآتية إلى A و B و C :

$$A \leftarrow 38.6, \quad B \leftarrow 42.4, \quad C \leftarrow 56.1$$

سطر ٦ يجمع الحاسب قيم A و B و C وتخصص هذه القيمة (137.1) للمتغير P :
P ← 137.1

سطر ٧ يخص الحاسب P/2 للمتغير S :

$$S \leftarrow 68.55$$

سطر ٨ يوجد الحاسب قيمة التعبير الحسابي على اليمين ، حيث SQRT تعنى الجذر التربيعي ، ويخصص هذه القيمة
AREA إلى 817.56570 :

$$AREA \leftarrow 817.56570$$

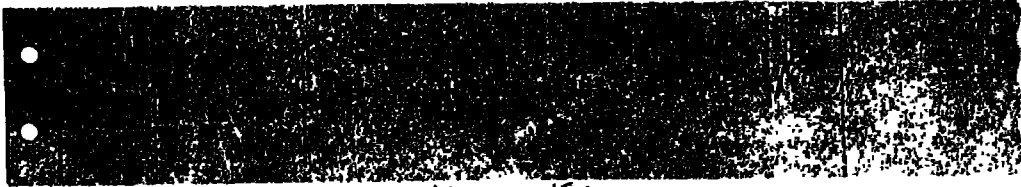
الأسطر ٩ - ١٠ يطبع الحاسب القيم A و B و C و P و AREA باستعمال حقول ذات عرض 7 = 1 - 8 ، 8 ، 8 ، 10 ، 12 على الترتيب كما في شكل ٣ - ٢ . لاحظ أن هذه الحقول تفصلها حقول خالية بعرض أربعة حروف (4X) لاحظ أيضاً أن P تم تقريبها إلى رقم عشري واحد (F10.1) وتم تقريب AREA إلى رقمين عشريين (F12.2) .

سطر ١١ تجبر الحاسب بأن يتوقف STOP .

سطر ١٢ تجبر جملة END المترجم بعدم وجود جبل أخرى في البرنامج تتطلب الترجمة ولن تكون في البرنامج وقت التنفيذ .

8-1=7	4	8	4	8	4	10	4	12				
1		2		3		4		5		6		7

1234567890123456789012345678901234567890123456789012345678901234567890123456789012345



شكل ٣ - ١٢

ملحوظة: عند تنفيذ البرنامج السابق، أشرنا إلى أن 38.6 خصصت إلى A. والرقم الحقيقي الذى تم تخزينه في A هو 0.38600000E+02 . حيث أن التمثيل الداخلى للقيم الحقيقية تكون في الشكل الأسى بأرقام عشرية تتراوح ما بين 8 إلى 9 تقريباً (انظر ملحق أ) .

مسائل محلولة

إدخال / إخراج غير مناسب

١ - ٣ اكتشف الأخطاء، إن وجدت ، في جمل I/O التالية غير المصاغة

READ, ID, WAGE, MAX, MIN (ج) READ A, B, X, Y (أ)

PRINT, ROOT, COEF, RANGE (د) PRINT NUMB, INT. RATE (ب)

(أ) يجب أن تكون هناك فصلة بعد READ .

(ب) يجب أن تكون هناك فصلة بعد PRINT ويجب ألا توجد فصلة بعد RATE .

(ج) صواب

(د) يجب ألا توجد فصلة بعد RANGE .

٢ - ٣ افترض تثقيب ثلاث بطاقات بيانات كالتالي :

أول بطاقة : 22.2, 3.33, 444

ثاني بطاقة : 555, 666, 77.77

ثالث بطاقة : 8.888, 9.99

أوجد القيم المخصصة للمتغيرات عند تنفيذ الآتي :

READ, A, B, J, K (ب)
READ, X, YREAD A, B (أ)
READ, J, K, X, Y

(أ) عندما ينفذ الحاسب أول جملة READ يخصص 22.2 إلى A و 33.3 إلى B وبذلك يتم تنفيذ أول جملة READ وتهمل البطاقة الأولى رغم أن القيمة الثالثة لم تقرأ بعد . وعندما ينفذ الحاسب جملة READ الثانية ، سيقراً من البطاقة الموجودة على قمة مجموعة البطاقات ، بطاقة البيانات الثانية . ومن ثم سيخصص 555 إلى J ، 666 إلى K ، 77.77 إلى X ثم يقرأ البطاقة التالية ليخصص 8.888 إلى Y وبذلك ، نحصل على :

A = 22.2, B = 3.33, J = 555, K = 666, X = 77.77, Y = 8.888

(ب) تعطى جملة READ الأولى :

A = 22.2, B = 3.33, J = 444, K = 555

وتعطى جملة READ الثانية :

X = 8.888, Y = 9.99

نلاحظ أنه تم إهمال بطاقة البيانات الثانية بعد تنفيذ جملة READ الأولى . وبذلك استخدمت بطاقة البيانات الثالثة لتنفيذ جملة READ الثانية .

٣ - ٣ افترض أن المطلوب تخصيص القيم 123 ، 456 ، 7.77 ، 8.88 و 9.99 للمتغيرات L ، M ، X ، Y و Z على الترتيب . اكتب جملة READ غير مصاغة ووضح كيف يجب أن تثقب البيانات على بطاقات البيانات إذا :
(أ) استخدمت بطاقة بيانات واحدة فقط ، (ب) استخدمت بطاقتان للبيانات القيم الصحيحة على البطاقة الأولى والقيم الحقيقية على البطاقة الثانية .

في كلا الحالتين لدينا جملة READ التالية :

READ, L, M, X, Y, Z

يجب أن تثقب البيانات كالتالي :

(أ) أول بطاقة : 123, 456, 7.77, 8.88, 9.99

(ب) أول بطاقة : 123, 456

ثاني بطاقة : 7.77, 8.88, 9.99

٣ - ٤ اعتبر أنه تم تثقيب بطاقة بيانات كالتالي :

بطاقة بيانات : 123, 44.4

(أ) حدد المخرج عند تنفيذ جزء البرنامج التالي :

```
READ, J, X
K = J**2
Y = 3.0*X
PRINT, J, K, Y
```

(ب) اشرح الفرق إذا استبدلت جملة PRINT الوحيدة السابقة بالجمليتين التاليتين

```
PRINT, J, K
PRINT, Y
```

(أ) تنفيذ البرنامج :

سطر ١ تخصص قيم بطاقة البيانات إلى J و X وينتج عن ذلك أن $J = 123$ ، $X = 44.4$

سطر ٢ تخصص مربع J إلى K وبهذا تكون قيمة $K \leftarrow 15129$.

سطر ٣ يخصص ثلاثة أضعاف قيمة X إلى Y وبهذا تكون قيمة $Y = 133.2$

سطر ٤ تطبع قيم J و K و Y مع قيمة Y في الشكل الأسى القياسى . وبذلك يظهر المخرج كالتالي :

123 15129 0.1332E 03

(ب) ستظهر قيم J و K على سطر واحد وقيمة Y على السطر الثاني وبذلك يظهر المخرج كالتالي :

123 15129
0.1332E 03

الإدخال المصاغ

٣ - ٥ حدد موقع الحلاً ، إن وجد ، في كل جملة من جمل READ وجملة FORMAT المصاحبة لها .

11 READ(5, 11), A, B, C, D, J, K, L (ج) READ(5, 11) A, K, M, Z, (أ)
11 FORMAT(4F15.2, 3I15) 11 FORMAT(F8.0, I15, I10, I15)
11 READ(5, 11) A, B, J, K, L (ب)
11 FORMAT(3F8.1, 2I8)

(أ) أولاً ، لا يجب أن تكون هناك فصلة بعد Z . ثانياً ، المتغير الرابع في جملة READ هو المتغير الحقيقي Z ولكن المواصفات المناظرة له في جملة FORMAT هي الحقل الصحيح I 15

(ب) لاحظ أولاً أن جملة FORMAT هو اختصار للجملة

11 FORMAT(F8.1, F8.1, F8.1, I8, I8)

وبذلك فالمتغير الصحيح J أعطى مواصفات حقيقية F8.1 .

(ج) أولاً ، يجب ألا تكون هناك فصلة قبل A . ثانياً ، مجموع عرض الحقول هو 105 وهو كبير جداً على بطاقة البيانات التي بها 80 عمود فقط .

٣ - ٦ افترض أنه تم تتبع بطاقة بيانات كما في الشكل ٣ - ١٣ . أوجد قيم J و K و L كنتيجة لكل من الحالات الآتية:

13 READ(5, 13) J, K, L (ج) 11 READ(5, 11) J, K, L (أ)
13 FORMAT(2I6, I2) 11 FORMAT(I4, I3, I8)
12 READ(5, 12) J, K, L (ب)
12 FORMAT(I3, 2X, I6, 3X, I3)

1	2
12345678901234567890	1234567890
1234567	33333

شكل ٣ - ١٣

لاحظ أولاً أن الأعداد الصحيحة فقط هي التي تقرأ كتغيرات حيث أن كل مواصفات الحقول الرقمية هي في الصيغة Iw حيث w هي عرض الحقل .

(أ) عرض الحقول هو 4 و 3 و 8 على الترتيب . من ثم ، ينحصر إلى L العدد الصحيح المثقب في أول أربعة أعمدة (الأعمدة 1 إلى 4) وينحصر إلى K العدد الصحيح المثقب في الأعمدة الثلاثة التالية (الأعمدة 5 إلى 7) وينحصر إلى L العدد الصحيح المثقب في الأعمدة الثمانية التالية (الأعمدة 8 إلى 15) وبذلك تصحیح :

$$L = 3333 \text{ و } K = 567 \text{ و } J = 1234$$

(ب) تأمر مواصفات الحقول الخمسة الحاسب بالتالي :

- 13 خصص إذ J العدد الصحيح المثقب في أول ثلاثة أعمدة (الأعمدة 1 إلى 3) : لذا تخصص 123 إلى J .
 2X اترك العمودين التاليين (الأعمدة 4 إلى 5) ، حتى إذا كانت هناك معلومات مثقبة في الأعمدة .
- 16 خصص العدد الصحيح المثقب في الأعمدة الستة التالية (أعمدة 6 إلى 11) إلى K حيث أن المسافات الخالية تفسر كأصفار في الحقول الرقمية . 670003 تخصص إلى K .
 3X اترك الأعمدة الثلاثة التالية (الأعمدة 12 إلى 14) .
- 13 خصص العدد الصحيح المثقب في الأعمدة الثلاثة التالية (الأعمدة 15 إلى 17) إلى L وبذا تخصص 300 إلى L .
 وبذلك تصيح : $J = 123$ و $K = 670003$ و $L = 300$
- (ج) 2I6 ، تكرر 16 مرتين ؛ أي مواصفات الحقول هي 16 ، I6 ، I2 ، بمرض الحقول 6 و 6 و 2 على الترتيب .
 وينتج عن هذا .
 $J = 123456$ و $K = 700033$ و $L = 33$

٣ - ٧ افرض أن برنامجنا به الجمل الآتية :

READ(5, 21) A, B
 21 FORMAT(F10.2, F12.3)

أوجد القيم المخصصة لكل من A و B إذا قرأت البطاقة A من الشكل ٣ - ١٤ ، (ب) قرأت البطاقة B من الشكل ٣ - ١٤ .

	10	12
	1	2
	1234567890123456789012345	
بطاقة A	-12.345	67.89
بطاقة B	-12345	6789

شكل ٣ - ١٤

- لاحظ أولاً أن عرض الحقول هو 10 و 12 على الترتيب ولذا يخصص الرقم الحقيقي الموجود في الأعمدة 1 إلى 10 إلى A ويخصص الرقم الحقيقي الموجود في الأعمدة 11 إلى 22 إلى B .
- (أ) وحيث أن 12.345 — مثقبة في الحقل الأول (الأعمدة 1 إلى 10) فهي تخصص إلى A وبما أن 67.89 مثقبة في الحقل الثاني (الأعمدة 11 إلى 22) فهي تخصص إلى B . حيث أن الرقنين لهما علامات عشرية فيمكن أن تحمل 2 في F10.2 وتحمل 3 في F12.3 .
- (ب) لاحظ عدم وجود علامة عشرية مثقبة في الحقل الأول (الأعمدة 1 إلى 10) لذلك ستسبب مواصفات الحقل F10.2 في إضافة علامة عشرية بعد مكانين من الجانب الأيمن للحقل . أي ما بين العمود 8 والعمود 9 . ومن ثم تخصص

123.45 — إلى A . أيضاً العدد المثقب في الحقل الثاني (الأعمدة 11 إلى 22) ليس به علامة عشرية . لذا فالموصفات F12.3 ، تضيف علامة عشرية بعد ثلاثة أماكن من الجانب الأيمن للحقل . أى ما بين العمود 19 والعمود 20 . ومن ثم ، تخصصر 678.9 إلى B . وأتينا نؤكد أن موضع الرقم في الحقل الخاص به سيختلف إذا تثقب بدون علامة عشرية

٣ - ٨ افترض أن برنامجاً به الجمل الآتية :

READ(5, 22) A, B
22 FORMAT(E12.4, E15.6)

أوجد القيم المخصصة لـ A و B إذا كانت بطاقة البيانات (أ) بطاقة (1) من الشكل ٣ - ١٥ ، (ب) بطاقة رقم ١ من الشكل ٣ - ١٥ .

	12				15		
	1		2		3		
	12345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901
بطاقة (1)	72.53E-04		1234567E+2				
بطاقة (2)	123456 E+3		5.28 E+2				

شكل ٣ - ١٥

لاحظ أن عرض الحقول هو 12 و 15 على الترتيب ؛ من ثم ، يخصص الرقم الحقيقي الموجود في الأعمدة 1 إلى 12 إلى A ويخصص الرقم الحقيقي الموجود في الـ 15 عموداً التالية ، أى (الأعمدة 13 إلى 27) إلى B .

(أ) لاحظ أن 75.23E-04 مثقب في الحقل الأول (الأعمدة 1 إلى 12) ، ولذا تخصصر إلى A . تهمل 4 و مواصفات الحقل E12.4 حيث أن الرقم مثقب بعلامة عشرية . وفي الحقل الثاني (الأعمدة 13 إلى 27) ولأن الرقم مثقب بدون علامة عشرية لذا تدل 6 في مواصفات الحقل E15.6 الحاسب بأن يضيف علامة عشرية بعد ستة أماكن من يسار الحرف E أى ما بين العمود 8 و 9 . وبذلك تخصصر $1.23456 E + 2$ إلى B .

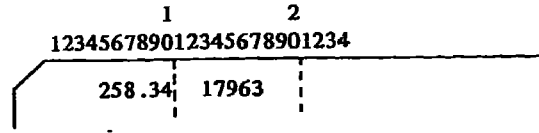
(ب) لاحظ أنه ليس هناك علامة عشرية مثقبة في الحقل الأول ، بالتالي ستسبب 4 و مواصفات الحقل E12.4 في إضافة علامة عشرية ما بين العمود الرابع والخامس إلى يسار الحرف E ، أى ما بين العمود 5 و 6 . من ثم تخصصر $12345.6 E + 3$ إلى A . وحيث أن الرقم المثقب في الحقل الثاني به علامة عشرية لذلك تعامل المسافة الخالية في العمود 27 كصفر . لذلك تخصصر $5.28 E + 20$ (وليس $5.28 E + 2$) إلى B .

٣ - ٩ افترض أن بطاقة بيانات هي كما في الشكل ٣ - ١٦ . أوجد قيم المتغيرات كنتيجة لكل من أزواج الجمل READ — FORMAT الآتية :

READ(5, 23) A, B
23 FORMAT(E10.3, E10.4) (ب)

READ(5, 21) A, K
21 FORMAT(F10.1, I10) (أ)

READ(5, 22) A, B
22 FORMAT(F10.3, F10.6) (ب)



شكل ٣ - ١٦

(أ) بطاقة المواصفات تبين أن عرض الحقلين 10 و 10 على الترتيب وتأمّر الحاسب بالآتي :
 F10.1 تخصص الرقم الحقيقي المثقب في أول 10 أعمدة (الأعمدة 1 إلى 10) إلى A لذا تصبح $A = 258.34$
 وتهمل 1 في F10.1 حيث أن الرقم المثقب في الحقل به علامة عشرية .

I 10 يخصص العدد الصحيح المثقب في 10 أعمدة التالية (الأعمدة 11 إلى 20) إلى K حيث تقرأ المسافات
 الخالية كأصفار ، لذا تصبح $K = 17963000$

(ب) كما في الجزء (أ) ، نحصل على $A = 258.34$. تخبر مواصفات الحقل الثاني الحاسب أن يخصص الرقم الحقيقي
 في الأعمدة 11 إلى 20 إلى B حيث لا توجد علامة عشرية بداخل الحقل ، وتخبر 6 في F10.6 الحاسب بإضافة
 علامة عشرية على بعد ست مسافات من يسار العمود الأخير (عمود 20) . لذا نحصل على $B = 17.963$

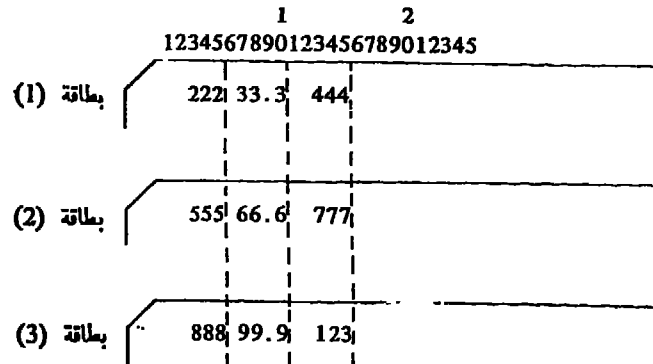
(ج) لاحظ أولاً أن حرف E لا يظهر على بطاقة البيانات ، لذا تماثل مواصفات الحقول *Ewd* نفس معاملة مواصفات
 الحقول *Fwd* . تحتوى مواصفات الحقلين على عرض حقول 10 ، 10 على الترتيب وتأمّر الحاسب بالآتي :
 E10.3 تخصص الرقم الحقيقي المثقب في أول 10 أعمدة (الأعمدة 1 إلى 10) إلى A وحيث أن الحقل به علامة
 عشرية ، فتخصص 258.34 إلى A .

E10.4 تخصص الرقم الحقيقي المثقب في الأعمدة العشرة التالية (الأعمدة 11 إلى 20) إلى B حيث توجد علامة
 عشرية بداخل الحقل. تضاف علامة عشرية على بعد 4 مسافات من يسار العمود الأخير . لذا تصبح قيمة $B = 1796.3$

٣ - ١٠ افترض أنه تم تثقيب البطاقات الثلاث الأولى من مجموعة بطاقات البيانات كما في الشكل (٣-١٧) . أوجد قيم المتغيرات
 لكل من الحالات الآتية :

(ب) READ(5, 12) J, K, A, L, B, M
 12 FORMAT(I5/I5, F5.2/I5, F5.2, I5)

(أ) READ(5, 11) J, A
 READ(5, 11) K, B, L, M
 11 FORMAT(I5, F5.2, I5)



شكل ٣ - ١٧

لاحظ أولاً أن كل مواصفات الحقول بمرص 5 .

(أ) تأنيذ أن جملة READ أو WRITE لا يكتمل تنفيذها إلا بعد يتم تخصيص كل القيم / طبع كل المتغيرات المذكورة . أول جملة READ تقرأ من البطاقة الأولى المتغيرات I و J وتقرأ بمواصفات جدول

IS و F5.2 على الترتيب . ويكتمل تنفيذ جملة READ هذه . وتترك البطاقة الأولى . تبدأ جملة READ الثانية أيضاً القراءة من البطاقة التي على القمة الآن ، أي البطاقة الثانية . حيث يوجد أربعة متغيرات وعددها أكثر من مواصفات الحقول (3) ، وبذلك تقرأ أكثر من سجال . في هذه الحالة ، تقرأ بطاقتان . تقرأ المتغيرات L ، B ، K من البطاقة الثانية باستعمال IS ، F5.2 ، IS على الترتيب لتصبح $K = 555$ ، $666 - L$. ثم يقرأ المتغير M من البطاقة 3 باستعمال IS ليصبح $M = 888$ وحيث قد تم تخصيص قيم لكل المتغيرات فقد اكتمل الآن تنفيذ جملة READ الثانية وتهمل البطاقة الثالثة .

(ب) حيث أن هناك ترمطين مائلتين ، فتقرأ على الأقل ثلاثة سجلات (بطاقات) . يقرأ المتغير J من البطاقة الأولى بـ IS وهكذا تصح $J = 222$. تقرأ المتغيرات A و K من البطاقة الثانية بـ IS ، F4.2 على الترتيب ، وهكذا $K = 555$ ، $A = 66.6$ وأخيراً $L = 888$ ، $B = 99.9$ ، $M = 123$ من البطاقة الثالثة .

المخرجات المصاغة

٣ - ١١ استخراج الأخطاء ، إن وجدت في كل زوج WRITE-FORMAT :

WRITE(6, 32) J, Z, K, WRITE(6, 31), A, B, N
32 FORMAT(1X, I10, 4X, 2F10.3) (ب) 31 FORMAT(F10.2, 3X, I8, 5X, I6) (أ)

(اعتبر عدم وجود جملة نوع قد غيرت نوع أي من المتغيرات)

أ) لا يجب أن تكون هناك فصلة قبل A ثم إعطاء المتغير الحقيقي B مواصفات الحقل الصحيح I8 .

ب) لا يجب أن تكون هناك فصلة بعد K . أيضاً تم إعطاء المتغير الصحيح K مواصفات الحقل الحقيقي F10.3 .

٣ - ١٢ افرض أن K تحتوي 12345 وتم تنفيذ جملة WRITE التالية :

WRITE(6, 41) K

صف المخرج إذا كانت جملة FORMAT المصاحبة هي :

41 FORMAT(I10) (أ) 41 FORMAT(4X, I3) (ج)

41 FORMAT(4X, I8) (ب) 41 FORMAT(I5) (د)

(أ) حيث أن عرض الحقل 10 بذلك ستطبع قيمة مضبوطة من الطرف الأيمن في أول (9 = 10 - 1) تسع أعمدة من صفحة الطباعة ، كما في الشكل ٣-١٨ (أ) . بمعنى آخر تتولد قيمة K مضبوطة من الطرف الأيمن في أول 10 حروف من سجل الإخراج ، أي تتولد bbbbb12345 لكن تستعمل أول مسافة ، للتحكم في العربة ولا تطبع .

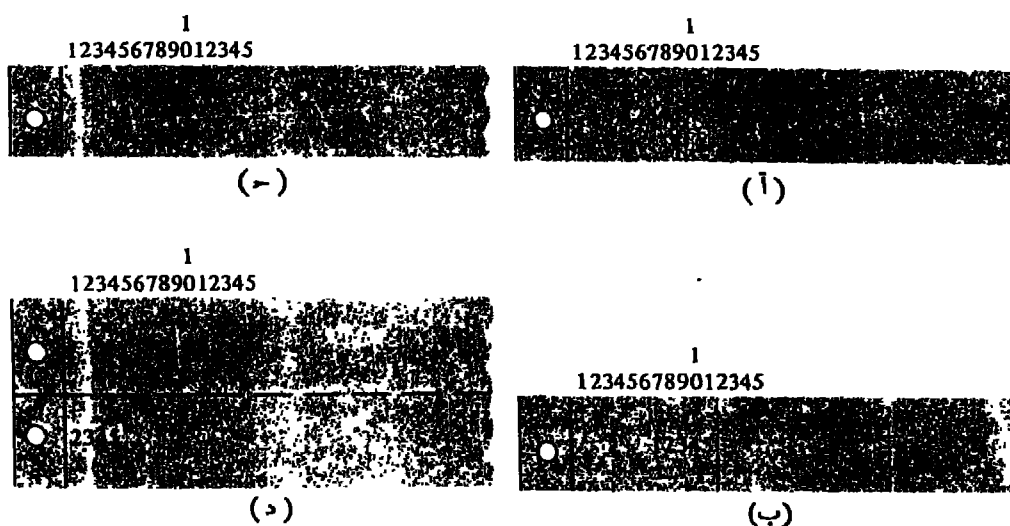
(ب) سيظهر المخرج كما في الشكل ٣ - ١٨ (ب) حيث تأمر مواصفات الحقول الحاسب بالآتي :

4X تقدم سطرًا واطبع (3 = 1 - 4) ثلاث مسافات في أول ثلاثة أعمدة من صفحة الطباعة .
 I8 اطبع قيمة K مضبوطة من الطرف الأيمن في الـ 8 أعمدة التالية (الأعمدة 4 إلى 11) بمعنى آخر ، تتولد أربع مسافات bbbb و bbb12345 على الترتيب . في سجل الإخراج ، ولكن يستعمل أول مسافة للتحكم في العربة ولا تطبع .

(ج) تأمر مواصفات الحقول الحاسب بالآتي :

4X تقدم سطرًا واحدًا واطبع (3 = 1 - 4) ثلاث مسافات في أول ثلاثة أعمدة من صفحة الطباعة .
 I3 اطبع قيمة K مضبوطة من الطرف الأيمن في الأعمدة الثلاثة التالية (الأعمدة 4 إلى 6) .
 حيث أن عرض الحقل المخصص لـ K أقل من عدد الخانات في K ، إما أن تطبع ثلاث نجوم *** كما في الشكل ٣ - ١٨ ، أو نبتّر قيمة K من اليسار ، وبذلك تطبع 345 فقط .

(د) عرض الحقل I5 مساو لعدد الخانات في K لذا تتولد 12345 في سجل الإخراج . ولكن يستعمل أول حرف «I» للتحكم في العربة وتجنّب وحدة الطباعة بأن تتقدم إلى قمة الصفحة التالية قبل طباعة الحروف الأربعة الباقية في أول (4 = 1 - 5) أربعة أعمدة من صفحة الطباعة كما في الشكل ٣ - ١٨ (د) .



شكل ٣ - ١٨

٣ - ١٣ افترض أن A تحتوي 135.2837 وتم تنفيذ جملة WRITE التالية :

WRITE(6, 42) A

صف المخرج إذا كانت جملة FORMAT المصاحبة هي :

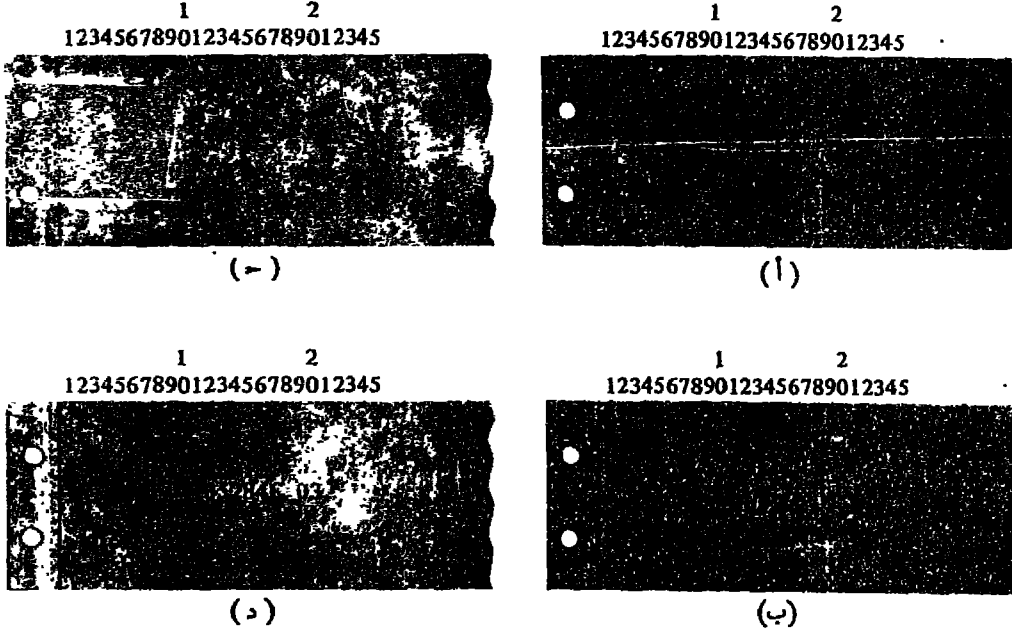
42 FORMAT(6X, F6.3) (ج)

42 FORMAT(6X, F10.1) (أ)

42 FORMAT(6X, E15.6) (د)

42 FORMAT(6X, F12.2) (ب)

في كل حالة ، تسبب مواصفات الحقل 6X تقدم وحدة الطباعة سطرًا واحدًا وأن تطبع مسافات في أول (5 - 1 = 6) أعمدة (الأعمدة 1 إلى 5) من صفحة الطباعة . يظهر المخرج كما في الشكل ٣ - ١٩ .



شكل ٣ - ١٩

وتأمر مواصفات الحقل الثاني الحاسب بالآتي :

- (أ) تقريب قيمة A إلى مكان عشري واحد ، أي ، إلى 135.3 وطباعة هذا الرقم مضبوطاً من الطرف الأيمن في الأعمدة الستة التالية (الأعمدة 6 إلى 15) .
- (ب) تقريب قيمة A إلى مكانين عشريين ، أي ، إلى 135.28 وطباعة هذا الرقم مضبوطاً من الطرف الأيمن في الإثنى عشر عموداً التالية (الأعمدة 6 إلى 17) .
- (ج) تقريب قيمة A إلى ثلاثة أماكن عشرية ، أي ، إلى 135.284 وطباعة هذا الرقم مضبوطاً من الطرف الأيمن في الأعمدة الستة التالية (الأعمدة 6 إلى 11) وحيث أن عرض الحقل (6) أقل من عدد الحروف (7) المراد طباعتها فإما تطبع ستة نجوم في الحقل أو يبتدئ الرقم من اليسار بحيث تطبع 35.284 في الحقل .
- (د) تقريب قيمة A إلى ستة أرقام معنوية ، أي 135.284 وطباعة هذه القيمة في الشكل الأسى القياسي ، أي 0.135284E03 مضبوطاً من الطرف الأيمن في الخمسة عشر عموداً التالية (الأعمدة 6 إلى 20) .

٣ - ١٤ افترض ID = 112233 و WAGE = 275.5 . حدد المخرج لكل زوج WRITE-FORMAT :

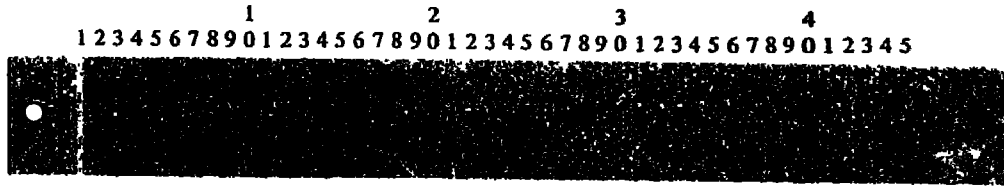
- (أ) WRITE(6, 11) ID
11 FORMAT(6X, 'IDENTIFICATION NUMBER', 2X, I8)
WRITE(6, 12) WAGE
12 FORMAT(6X, 'WEEKLY SALARY \$', F8.2)
- (ب)

(أ) تتقدم وحدة الطباعة سطرًا واحدًا وتطبع مسافات في أول (5 = 1—6) خمسة أعمدة (الأعمدة 1 إلى 5) من صفحة الطباعة . يتسبب المدخل

'IDENTIFICATION NUMBER'

في طباعة الرسالة بين الفواصل العليا في الحقل التالي . حيث أن الرسالة تشغل 21 مسافة ، ستطبع في عدد 21 عموداً التالية (الأعمدة 6 إلى 26) - تتسبب 2X في طبع مسافات خالية في الأعمدة 27 و 28 ، وتتسبب 18 في طباعة قيمة ID مضبطة من الطرف الأيمن في الأعمدة الثمانية التالية (الأعمدة 29 و 36) . أنظر شكل ٣ - ٢٠ (أ) .

(ب) يظهر المخرج كما في الشكل ٣ - ٢٠ (ب) .



(أ)



(ب)

شكل ٣ - ٢٠

٣ - ١٥ افرض $B = 0.00285293$ ، $A = 256.174$ ، $M = 138$ ، $D = 25367$ أوجد المخرج عند تنفيذ الزوج WRIE-FORMAT التالي :

WRITE(6, 12) ID, A, B (ب)

12 FORMAT('1', I10, 2E15.5)

WRITE(6, 11) ID, M, A, B

11 FORMAT(' ', I10, I6, F8.1, F10.3) (أ)

(أ) أول رمز في سجل الإخراج هو مسافة معطاة بـ « » تمثل إشارة التحكم في العربة ، ومن ثم يتقدم سطرًا واحدًا . مواصفات الحقول الأربعة الباقية عرضها 10 و 6 و 8 و 10 على الترتيب ، وتأمّر الحاسب بالآتي :

I10 . وضع القيمة الصحيحة ID مضبطة من الطرف الأيمن في أول عشرة أعمدة من صفحة الطباعة .

I6 طباعة قيمة M مضبطة من الطرف الأيمن في الأعمدة الستة التالية (الأعمدة 11 إلى 16) .

F8.1 تقريب قيمة A إلى مكان عشري واحد أي 256.2 ، وطباعة هذا الرقم الحقيقي مضبلاً من الطرف الأيمن في الأعمدة الثمانية التالية (الأعمدة 17 إلى 24) .

F10.3 تقريب قيمة B أي 0.003 - إلى 3 أماكن عشرية ، وطباعة هذا الرقم الحقيقي مضبلاً من الطرف الأيمن في الأعمدة العشرة التالية (الأعمدة 25 إلى 34) .

ومن ثم يظهر المخرج كما في الشكل ٣ - ٢١ (أ) .

(ب) تأمر مواصفات الحقل الحاسب بالآتي :

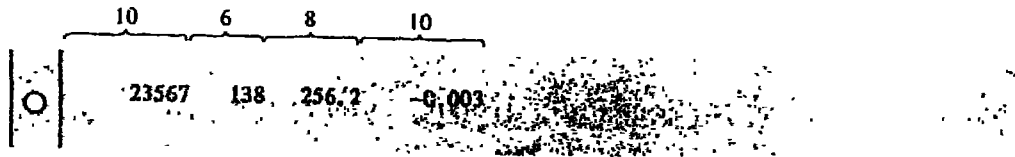
١٠ - أمر التحكم و المرية للتقدم إلى صفحة جديدة .

110 طاعة قيمة ID في أول عشرة أعمدة من صفحة الطباعة .

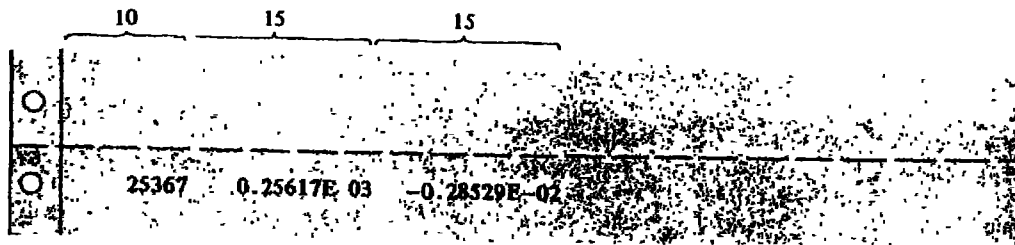
E15.5 تقريب قيمة المتغير A إلى 5 أرقام معنوية ، أي 256.17 وطباعة هذه القيمة في الشكل الأسى القياسى . أى
0.25617E03 مضبطاً من الطرف الأيمن في الخمسة عشر عموداً التالية (الأعمدة 11 إلى 25) .

E15.5 تقريب قيمة B إلى 5 أماكن معنوية ، أى 0.0028529 - وطباعة هذا الرقم في الشكل الأسى القياسى . أى
0.28529E-02 مضبطاً من الطرف الأيمن في الخمسة عشر عموداً التالية (الأعمدة 26 - 40) .

يظهر المخرج كما في الشكل ٢ - ٢١ (ب)



(أ)



(ب)

شكل ٢ - ٢١

٢ - ١٦ افرض أن المطلوب طباعة المتغيرات J ، A ، K و B بحيث يخصص لكل متغير صحيح 4 أعمدة ويخصص لكل رقم - 8 أعمدة بثلاثة أماكن عشرية وتفصل القيم عن بعضها بثلاث مسافات على الأقل . أوجد مدى القيم التي يمكن أن تطبع لتر متغير واكتب زوج WRITE-FORMAT الذى (أ) يطبع القيم على سطر واحد (ب) يطبع قيم J و A على سطر و K و B على سطر آخر .

يجب أن تقع قسم J و K بين 999- و 9999 و يجب أن تقع قسم A و B بين 999.999- و 9999.999

(أ) WRITE(6, 11) J, A, K, B
11 FORMAT(1X, I4, 3X, F8.3, 3X, I4, 3X, F8.3)
or
11 FORMAT(1X, 2(I4, 3X, F8.3, 3X))

٢ - ١٧ افرض أن J3 ، J2 ، J1 تحتوى على ثلاث درجات مواد دراسية ، وتحتوى K على متوسطاتها .

(أ) اكتب جزء البرنامج الذى سيطبع الآتى على صفحة جديدة مبتدأ من عمود 11 :

TEST SCORES
XXX
XXX
XXX
AVERAGE
XXX

حيث تشير XXX'S إلى قسم درجات المواد الدراسية ومتوسطاتها .

(ب) نفذ ماسبق بجملة WRITE واحدة

```

WRITE(6, 11)
11 FORMAT('1', 10X, 'TEST SCORES')
WRITE(6, 12) J1, J2, J3
12 FORMAT(1X, I17)
WRITE(6, 13)
13 FORMAT(13X, 'AVERAGE')
WRITE(6, 12) K
WRITE(6, 21) J1, J2, J3, K
21 FORMAT('1', 10X, 'TEST SCORES'/1X, I17/1X, I17/1X, I17/13X, 'AVERAGE'/1X, I17)
or
21 FORMAT('1', 10X, 'TEST SCORES'/3(1X, I17//)13X, 'AVERAGE'/1X, I17)

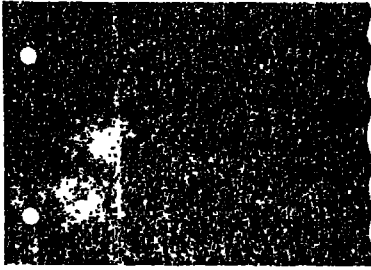
```

٣ - ١٨. عرض $J1 = 111$ ، $J2 = 222$ ، $J3 = 333$ ، $J4 = 444$ ، $J5 = 555$ حدد الخرج إذا تم تنفيذ الزوج الآتي :

```

1
123456789012345

```



شكل ٣ - ٢٢

```

WRITE(6, 15) J1, J2, J3, J4, J5
15 FORMAT('0', I8/I9)

```

يستخدم الحاسب سطرين ، أي ، يتخطى سطر ('0') ويطيح
 J1 في أول 8 أعمدة بالمواصفات (I8). تعين الشرطة المائلة (/) عن نهاية
 السجل ، لذا تطبع J2 في أول (8 - 1 = 9) ثمانى أعمدة
 من السطر التالى بمواصفات I9 . تتكرر جملة FORMAT لـ J3
 و J4 وتتكرر مرة ثانية لـ J5 . تتسبب '0' في تخطى سطر بين
 J2 و J3 و بين J4 و J5 . سيظهر الخرج كما في الشكل ٣-٢٢

برامج بسيطة كاملة :

٣ - ١٩. تعطى المساحة السطحية (SUR) وحجم صندوق (VOL) ذا أبعاد a ، b ، c بالمعادلتين

$$VOL = abc \quad \text{و} \quad SUR = 2(ab + ac + bc)$$

افترض أن أبعاد الصندوق هي 23.5 ، 41.3 ، 16.2 على الترتيب وتم تثقيبها على بطاقة بيانات كما في الشكل ٣ - ٢٣
 (أ) أوجد الخرج للبرنامج التالى :

```

C
C SURFACE AREA AND VOLUME OF A BOX
C
READ(5, 61) A, B, C
61 FORMAT(3F10.2)
SUR = 2.0(A*B + A*C + B*C)
VOL = A*B*C
WRITE(6, 62) A, B, C, SUR, VOL
62 FORMAT(6X, 'DIMENSIONS OF BOX'/7X, 3(F5.1, 1X)//
1 6X, 'SURFACE AREA =', F10.1//6X, 'VOLUME =', F16.2)
STOP
END

```

1	2	3
12345678901234567890123456789012345		
23.5	41.3	16.2

(أ)

1	2	3
12345678901234567890123456789012345		
[REDACTED]		

(ب)

شكل ٣ - ١٣

تنفيذ البرنامج

يقرأ الحاسب قسم A ، B ، C تبعاً لجملة FORMAT المرقمة 61 والتي هي اختصار للاق :

61 FORMAT(F10.2, F10.2, F10.2)

في كل حالة ، عرض الحقل هو 10 لذا تخصص الأرقام المتبقية في (الأعمدة 1 إلى 10 ، 11 إلى 20 ، 21 إلى 30) إلى A و B و C على الترتيب . وهذا تصبح $A = 23.5$ ، $B = 41.3$ ، $C = 16.2$ (تعمل 2 في 10.2 حيث أن الأرقام على بطاقة البيانات بها علامات عشرية فعلاً) .

يحسب الحاسب $2.0 * (A * B + A * C + B * C)$ وهي 4040.62 ويخصص هذا الرقم إلى SUR ويحسب $A * B * C$ وهي 15722.910 ويخصص هذا الرقم إلى VOL :

$SUR \leftarrow 4040.62$ ، $VOL \leftarrow 15722.910$

يأمر الزوج WRITE-FORMAT الحاسب بطباعة المخرج كما في الشكل ٣-٢٣ (ب) . لاحظ أن شرطة مائلة واحدة / تسبب تقدم وحدة الطباعة إلى السطر التالي ، ولكن شرطتان مائلتان تسببان تقدم وحدة الطباعة سطران ، ومن ثم ، تتخطى سطرًا ، لاحظ أيضاً أن SUR تم تقريبها إلى مكان عشري واحد (F10.1) ، وVOL تم تقريبها إلى مكانين عشريين (F16.2) .

٣ - ٢٠ . بافتراض أن $D = a_1b_2 - a_2b_1 \neq 0$ ، فإن حل المعادلتين الخطيتين

$$a_1x + b_1y = c_1$$

$$a_2x + b_2y = c_2$$

يعطى بواسطة

$$x = \frac{b_2c_1 - b_1c_2}{a_1b_2 - a_2b_1} \quad y = \frac{a_1c_2 - a_2c_1}{a_1b_2 - a_2b_1}$$

اكتب برنامج فورتران يقرأ قسيم A1 و B1 و C1 من بطاقة بيانات واحدة ثم قيم A2 و B2 و C2 من بطاقة بيانات ثانية :
أحسب القيمتين X ، Y ثم اطبع المخرج حتى يظهر كالاتي :

COEFFICIENTS OF LINEAR EQUATIONS

FIRST *****

SECOND *****

SOLUTION OF LINEAR EQUATIONS

X = *****

Y = *****

C

C

C

SOLUTION OF LINEAR EQUATIONS

READ(5, 11) A1, B1, C1, A2, B2, C2

11 FORMAT(3F10.1)

D = A1*B2 - A2*B1

X = (B2*C1 - B1*C2)/D

Y = (A1*C2 - A2*C1)/D

WRITE(6, 12) A1, B1, C1, A2, B2, C2, X, Y

12 FORMAT(11X, 'COEFFICIENTS OF LINEAR EQUATIONS'//15X, 'FIRST',

1 3X, 3F8.1/15X, 'SECOND', 2X, 3F8.1//

1 11X, SOLUTION OF LINEAR EQUATIONS'//15X, 'X =',

1 2X, F12.2/15X, 'Y =', 2X, F12.2)

STOP

END

مسائل تكميلية

إدخال / إخراج غير مصاغ

٣ - ٢١ اكتشف الأخطاء ، أن وجدت ، في جمل I/O غير المصاغه

READ, INT, LOT, AREA (ج) READ, FIRST, LAST, NEXT, (١)
PRINT, A, B, C, D, (د) PRINT ID, WAGE, RATE, (ب)

٣ - ٢٢ افرض أن أول أربع بطاقات من مجموعة بطاقات البيانات تم تثقيها كالاتي :

أول بطاقة : 4.4 ، 3.3 ، 22 ، 11

ثاني بطاقة : 5.5 ، 6.6 ، 77

ثالث بطاقة : 88 ، 99 ، 2.34

رابع بطاقة : 5.67 ، 8.90 ، 123 ، 456

أوجد القسم المخصصة المتغيرات لو نفذنا الآتي :

READ, J, K, A (-)	READ, J, K, A (ب)	READ, J, K, A, B, C (أ)
READ, B, C, L	READ, B, C, L, M	READ, L, M, N, Y
READ, M	READ, X, Y	•
READ, X, Y		

٢ - ٢٣ افرض أن القسم 1.1 و 2.2 و 33 و 44 و 55 سوف يتم تخصيصها إلى المتغيرات A و B و J و K و L على الترتيب ،
ويستعمل جملة READ التالية :

READ, A, B, J, K, L

بين كيف يجب أن تتقب البيانات على البطاقات إذا (أ) استعملت بطاقة واحدة فقط (ب) استعملت بطاقتان للبيانات مع وجود القيم الحقيقية على البطاقة الأولى والقسم الصحيحة على البطاقة الثانية .

٣ - ٢٤ افرض أنه تم تتقيب بطاقتين للبيانات كالآتي :

أول بطاقة : 22.2 و 15

ثاني بطاقة : 25 و 35

أوجد المخرج إذا نفذ جزء البرنامج التالي :

READ, A, J, K (-)	READ, A, J, K (أ)
B = A*6.0	B = A*6.0
PRINT, A	PRINT, A, B, J, K
PRINT, B, J, K	
READ, A (د)	READ, A (ب)
READ, J, K	READ, J, K
B = A*6.0	B = A*6.0
PRINT, A, J	PRINT, A, B, J
PRINT, B	PRINT, K
PRINT, K	

الإدخال المصاغ

٣ - ٢٥ اكتشف الأخطاء ، إن وجدت في كل زوج READ-FORMAT .

30 READ(5, 30) S, T, U, J, K (-)	10 READ(5, 10) X, Y, L, M, (أ)
FORMAT(2F10.3, 3I8)	FORMAT(F8.1, F9.2, 2I12)
40 READ(5, 40) A, J, B, K, C, L (د)	20 READ(5, 20) A, L, B, M (ب)
FORMAT(F8.2, I6, F8.3)	FORMAT(F10.3, I8)

٢ - ٢٦ افرض أنه تم تثقيب بيانات كما في الشكل ٣ - ٢٤ ، وافرض أنه تم تنفيذ جملة READ الآتية :

READ(5, 50) J, K, L, M

أوجد القسم المخصصة لكل من J ، K ، L ، M إذا كانت جملة FORMAT المصاحبة هي :

- (ا) 50 FORMAT(I2, I3, I8, I3) (ب) 50 FORMAT(4I4)
 (ج) 50 FORMAT(2I3, 5X, 2I3) (د) 50 FORMAT(3I3, 2I4)

1					2					3				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	1	1	1	1	2	2	2	2	2	3	3	3	3	3

شكل ٣ - ٢٤

٣ - ٢٧ افرض أنه تم تنفيذ زوج READ-FORMAT الآتي :

READ(5, 10) A, B, C
 10 FORMAT(F10.2, F10.3, F10.2)

أوجد القيم المخصصة لكل من A و B و C إذا كانت بطاقة البيانات كما هو موضح في الشكل ٣ - ٢٥ .

1					2					3				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
11.11	-22.22	33.33			1111	-2222	3333			1111	-2.222	3333		

شكل ٣ - ٢٥

٣ - ٢٨ افرض أنه تم تنفيذ زوج READ-FORMAT الآتي :

READ(5, 20) A, B, C
 10 FORMAT(E10.2, E10.3, E10.2)

أوجد القيم المخصصة لكل من A و B و C إذا كانت بطاقة البيانات كما هو موضح في الشكل ٣ - ٢٦ .

	1	2	3
	1234567890	1234567890	123456789012345
(أ)	-44.4 E+2	5.55 E3	6.66E-04
(ب)	-444 E+2	555 E3	666E-04
(ج)	-44.4 E+2	555	6.66
(د)	-444E+2	5.55	666 E-04

شكل ٣ - ٢٦

٣ - ٢٩ افرض أنه تم تفتيق بطاقة بيانات كما في الشكل ٣ - ٢٧ . أوجد القيم المخصصة للمتغيرات إذا تم تنفيذ زوج READ-FORMAT الآتي :

40 READ(5, 40) R, S, T	(د)	10 READ(5, 10) I, J, A, B	(أ)
50 FORMAT(3F4.1)	(ج)	20 READ(5, 20) A, B, C, D	(ب)
60 READ(5, 60) I, J, K, X, Y, Z	(و)	30 READ(5, 30) J, K, X, Y, Z	(ح)
60 FORMAT(2X, 3I1, 3X, F4.2, 2F3.1)	(و)	30 FORMAT(I4, 2X, I4, 3F4.2)	(ح)

	1	2
	1234567890	123456789012345
	1234567890	1.11 222

شكل ٣ - ٢٧

٣ - ٣٠ افرض أنه تم تفتيق البطاقات الثلاث الأولى من مجموعة بيانات كما في الشكل ٣ - ٢٨ . أوجد القيم المخصصة للمتغيرات إذا تم تنفيذ الآتي :

40 READ(5, 40) J, X	(د)	10 READ(5, 10) J, X	(أ)
50 READ(5, 50) K, Y, Z	(ج)	20 READ(5, 20) J, A, B, K	(ب)
50 FORMAT(I5, 2F5.2)	(و)	30 READ(5, 30) J, A, B	(ح)
50 READ(5, 50) J, X, Y, Z, K, L, M	(و)	30 READ(5, 30) K, X, Y, L, M	(ح)
50 FORMAT(I5, 2F5.2/F5.2/I5, 5X, 2I4)	(و)	30 FORMAT(I5, 2F5.2, I5)	(ح)

	1					2				
	1	2	3	4	5	1	2	3	4	5
البطاقة الأولى	1	1	1	1	1	2	2	2	2	2
البطاقة الثانية	4	4	4	4	4	5	5	5	5	5
البطاقة الثالثة	7	7	7	7	7	8	8	8	8	8

شكل ٣ - ٢٨

نخرج المصاغ

٣ - ١ - اكتشاف الأخطاء ، أن وجدت ، في كل زوج WRITE-FORMAT

WRITE(6, 30), S, T, U, J, K (-) WRITE(6, 10) X, Y, M, N, (أ)
 30 FORMAT(2E12.3, 2(F8.1, 3X, I7)) 10 FORMAT(1X, 3F8.1, 2I9)
 WRITE(6, 40) A, J, B, C, K (د) WRITE(6, 20) A, L, B, M (ب)
 40 FORMAT(F7.1, I8, E6.2) 20 FORMAT(1X, F9.2, 2(I8, E7.3))

٣ - ٢ - افرض أن J و K و L تحتوي 1111 و 2222 و 3333 على الترتيب وتم تنفيذ

WRITE(6, 50) J, K, L

صف المخرج إذا كانت جملة FORMAT المصاحبة هي :

50 FORMAT(1X, I10) (د) 50 FORMAT(3I10) (أ)
 50 FORMAT(1X, 2I10) (هـ) 50 FORMAT(1X, I10/2I10) (ب)
 50 FORMAT(I4, 7X, I3, 7X, I4) (و) 50 FORMAT(1X, 'THE NUMBERS ARE', 3I8) (ج)

٣ - ٢ - افرض أن A و B و C تحتوي 111.222 و 444.666 و 777.888 على الترتيب ، وتم تنفيذ :

WRITE(6, 60) A, B, C

صف المخرج إذا كانت جملة FORMAT المصاحبة هي :

60 FORMAT(1X, 3F10.2) (أ)
 60 FORMAT(1X, F10.1, 2X, F5.3, 2X, F10.1) (ب)
 60 FORMAT('1', E15.5, E15.2, E15.4) (ج)
 60 FORMAT(1X, 2E12.4) (د)
 60 FORMAT(1X, F10.2///1X, E10.3///1X, F10.1) (هـ)
 60 FORMAT(F7.3, 2X, E15.7/F15.1) (و)

٣ - ٣٤ افرض أنه تم تنفيذ :

WRITE(6, 70) J, K, X, Y

(١)

مع فرض أن كل عدد صحيح سيخصص له خمسة أعمدة ، وأن كل رقم حقيقى سيخصص له سبعة أعمدة مع مكانين عشريين . سوف يفصل بين المتغيرات ثلاث مسافات على الأقل . أوحد مدى التيم التي يمكن أن تطبع لكل متغير . وأنتب جملة FORMAT المصاحبة التي :

(أ) تطبع القسم على سطر واحد

(ب) تطبع القسم على أول سطر لصفحة جديدة .

(ج) تطبع القيم على السطر الخامس لصفحة جديدة .

(د) تطبع الأعداد الصحيحة على سطر والأرقام الحقيقية على السطر التالي .

(هـ) تطبع كل رقم على سطر مختلف .

(و) تطبع الأعداد الصحيحة على ثالث سطر من صفحة جديدة والأرقام الحقيقية على السطر السادس من نفس الصفحة .

بمراجع :

٣ - ٣٥ افرض أن الضريبة الفيدرالية FTAX هي 15 في المائة من ضريبة الولاية STAX هي 2.5 في المائة من إجمال الدخل WAGE افرض بطاقة تحوى على رقم ID للموظف (9خانات) ثم على أجره في الساعة RATE وعدد الساعات HOUR التي اشتغلها خلال الأسبوع . اكتب برنامج يطبع الـ ID ، HOUR ، WAGE ، FTAX ، STAX والأجر الصافي للموظف TPAY .

٣ - ٣٦ افرض x و y و z هي الدوال الآتية في t .

$$x = t^3 - 8t + 4, \quad y = \sin t + \cos 2t, \quad z = e^{3t+5}$$

اكتب برنامجاً يقرأ t ، ويطبع t و x و y و z

٣ - ٣٧ افرض أن سيارة تحركت من سكون بمجلة ثابتة a لعدد من الثواني . وأن السرعة النهائية v والمسافة d التي سافرتها السيارة تعطى بالقوانين :

$$d = \frac{1}{2}at^2 \quad \text{and} \quad v = at$$

اكتب برنامجاً يقرأ a و t ويطبع t و d و v

٣ - ٣٨ اكتب برنامجاً يقرأ الارتفاع h وأطوال القاعدتين b_1 و b_2 لشبه منحرف T ويطبع h و b_1 و b_2 والمساحة T .

مسائل متنوعة :

٣ - ٣٩ افرض أن جملة READ الآتية تم تنفيذها :

READ(5, 10) A, B

أوجد عدد البطاقات المتخطاة قبل قراءة B إذا كانت جملة FORMAT هي :

10 FORMAT(F10.2////) (أ)
10 FORMAT(F10.2////F10.2) (ب)

[وبصورة عامة تتخطى عدد n من البطاقات إذا ظهر عدد n شروط ماثلة في نهاية جملة FORMAT كما في (أ) ،
ولكن تتخطى عدد $n - 1$ من البطاقات إذا ظهر عدد n من الشروط الماثلة في وسط جملة FORMAT كما في (ب)] .

٣ - ٤٠ افرض أنه تم تنفيذ جملة WRITE الآتية :

WRITE(5, 20) A, B

أوجد عدد الأسطر الخالية قبل طباعة B إذا كانت جملة FORMAT هي :

20 FORMAT(1X, F10.2////) (أ)
20 FORMAT(1X, F10.2////1X, F10.2) (ب)

[وبصورة عامة تطبع عدد $(n - 1)$ من الأسطر الخالية كلما ظهر عدد n من الشروط الماثلة في نهاية أو في وسط جملة
FORMAT (تارن مع مسألة ٣ - ٢٩)]

اجابات للمسائل التكميلية المختارة

٣ - ٢١ (أ) لا يجب أن تكون هناك فصلة بعد NEXT

(ب) يجب أن تكون هناك فصلة بعد PRINT ولا شيء بعد RATE.

(ج) لا يوجد أخطاء (صحيحة) .

(د) لا يجب أن تكون هناك فصلة بعد D .

٣ - ٢٢

11, 22, 3.3, 4.4, 5.5, 88, 99, 2.34, 5.67 (أ)
11, 22, 3.3, 5.5, 6.6, 77, 88, 5.67, 8.90 (ب)
11, 22, 3.3, 5.5, 6.6, 77, 88, 5.67, 8.90 (ج)

٣ - ٢٣ (أ) بطاقة البيانات : 1.1 و 2.2 و 33 و 44 و 55

(ب) أول بطاقة : 1.1 و 2.2

ثاني بطاقة : 33 و 44 و 55

- ٢٤ - ٣ (أ) 22.2 و 133.2 و 15 و 25 على سطر واحد .
 (ب) 22.2 و 133.2 و 15 على سطر و 25 على السطر التالي .
 (ج) 22.2 على سطر و 133.2 و 15 و 25 على السطر التالي .
 (د) 22.2 و 25 على سطر 133.2 على سطر تال ، ثم 35 على السطر التالي .
- ٢٥ - ٣ (أ) لا يجب أن تكون هناك فصلة قبل X ولا بعد M .
 (ب) لا يوجد أخطاء (صحيحة) .
 (ج) يقابل المتغير الحقيقي U رمز الشكل الصحيح 18 .
 (د) تتكرر جملة FORMAT K, J, C و L ولكن رموز الشكل ليست من النوع المناسب (الصحيح) .

٢٦ - ٣

- (أ) 111, 110, 222, 200
 (ب) 111, 110, 0, 222
 (ج) 11, 111, 222, 220
 (د) 1111, 1000, 22, 2220

٢٧ - ٣

- (أ) 11.11, -22.22, 33.33 (ب) 11110.00, -2.222, 333.30 (ج) 11.11, -2.222, 33.33

- (أ) -44.4E+2, 5.55E+3, 6.66E-4
 (ب) -4440.00E2, 5.550E3, 6.66E-4
 (ج) -44.4E20, 0.555, 6.66
 (د) -4.44E2, 5.55, 66.60E-4

- (أ) 12345, 67890, 1.11, 22.2 (ب) 12.345, 67.890, 1.11, 0.222 (ج) 1234, 7890, 1.1, 10.02, 22.00 (د) 123, 678.9, 1.0, 110, 222
 (أ) 12.345, 67.890, 1.11, 0.222 (ب) 123.4, 567.8, 900.1 (ج) 123.4, 567.8, 900.1 (د) 3, 4, 5, 90.01, 0.11, 0.2

- (أ) 11111, 22.2, 44444, 55.5, 66, 77777 (ب) 11111, 22.2, 0.33, 44444, 77777, 88.8 (ج) 11111, 22.2, 44444, 55.5, 0.00 (د) 11111, 22.2, 0.33, 444.44, 77777, 9, 9999
 (أ) 11111, 22.2, 0.33, 44444, 55.5, 0.66, 66666, 77777 (ب) 11111, 22.2, 0.33, 44444, 55.5, 0.66, 66666, 77777 (ج) 11111, 22.2, 0.33, 44444, 55.5, 0.66, 66666, 77777 (د) 11111, 22.2, 0.33, 444.44, 77777, 9, 9999

- ٣١ - ٣ (أ) لا يجب أن تكون هناك فصلة بعد N . يقابل المتغير الصحيح M رمز الشكل الحقيقي F8.1 .
 (ب) لا توجد أخطاء (صحيحة) .
 (ج) لا يجب أن تكون هناك فصلة قبل S . يقابل المتغير الصحيح K رمز الشكل الحقيقي F8.1 .
 (د) لا توجد أخطاء (صحيحة) .

$\text{bbbb1111 bbbbbb2222 bbbbbb3333}$ (أ) ٣٢ - ٣
 bbbb1111 (ب)
 $\text{bbbb2222 bbbbbb3333}$
 THE NUMBERS ARE $\text{bbbb1111 bbbb2222 bbbb3333}$. (ج)

(د) تظهر الأرقام على ثلاثة أسطر مختلفة .

(هـ) لا توجد إشارة صريحة للتحكم في العربة لذا يستعمل أول حرف من « 1111 » للتحكم في العربة ، وسيطج فقط الثلاثة الباقية . أيضاً ، عرض حقل K ليس كافياً ، لذا أما ستطبع 222 أو « . . . » وهكذا ، سيظهر واحد من الآتي على أول سطر من صفحة جديدة .

(٣) $111 \text{ bbbbbb} *** \text{ bbbbbb} 3333$ أو $111 \text{ bbbbbb} 222 \text{ bbbbbb} 3333$

$\text{bbbb111.22 bbbb444.67 bbbb777.89}$ (أ) ٣٢ - ٣
 $\text{bbbb111.2 bb} **** * \text{ bbbbbb777.9}$ (ب)
 على أول سطر من صفحة جديدة . $\text{bbbb0.11122E 03 bbbbbb0.44E 03 bbbbbb0.7779E 03}$ (ج)
 على السطر التالى . $\text{bb0.1112E 03 bb0.4447E 03, and then bb0.7779E 03}$ (د)
 على السطر السابع . bbbb777.9 ثم b0.445E03 على السطر الرابع ثم bbbb111.22 (هـ)
 على السطر bbbbbbb777.9 ثم $\text{11.222 bbbb0.4446660E 03}$ على السطر الأول من صفحة جديدة . ثم (و)

التالى .

٣ - ٣٤ يمكن أن تطبع ل و K لتقسيم بين 9999 - و 99999 ويمكن أن تطبع X و Y لتقسيم بين 999.99 - و 9999.99

FORMAT(1X, 2(I5, 3X), 2(F7.2, 3X)) (أ)
 FORMAT('1', 2(I5, 3X), 2(F7.2, 3X)) (ب)
 FORMAT('1'////1X, 2(I5, 3X), 2(F7.2, 3X)) (ج)
 FORMAT(1X, I5, 3X, I5/1X, F7.2, 3X, F7.2) (د)
 FORMAT(1X, I5/1X, I5/1X, F7.2/1X, F7.2) (هـ)
 FORMAT('1'////1X, I5, 3X, I5////1X, F7.2, 3X, F7.2) (و)

4 (ب) 5 (أ) ٣٩ - ٣

4 (ب) 4 (أ) ٤٠ - ٣

الفصل الرابع

نقل التحكم ، خرائط سير العمليات

٤ - ١ مقدمة

عادةً ينفذ الحاسب تعليمات برنامج الفورتران أمراً بعد الآخر إلا إذا أمر بغير ذلك . ويمكن التحكم في ترتيب تنفيذ الأوامر بعدة تعليمات مختلفة . أساساً يوجد نوعان من نقل التحكم ، انتقال غير مشروط وانتقال مشروط . وسوف ندرس جمل التحكم هذه في هذا الفصل ، والفصل الخامس ، والفصل الثامن عشر .

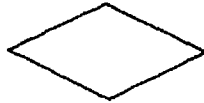
كلما تعقدت البرامج ، كلما أصبحت خريطة سير العمليات مساعداً كبيراً في التخطيط والتصميم ، وبناء هيكل البرنامج . وخريطة سير العمليات هي تمثيل بياني للخوارزم ، أي صورة مرئية لخطوات الخوارزم ، وأيضاً سير التحكم بين الخطوات المتعددة . (الخوارزم هي طريقة لحل المشكلة خطوة بخطوة) عموماً ، ففي خريطة سير العمليات نخطط كل عملية . أو أمر أو سلسلة من الأوامر بصندوق ، ونشير إلى سير التحكم بأنهم موجبة بين الصناديق . وعلاوة على ذلك ، فإننا نشير إلى أنواع العمليات المختلفة بأشكال مختلفة من الصناديق كما هو مبين في شكل ٤ - ١ . إذا استكلت خريطة سير العمليات في صفحة أخرى ، أو إذا كان من الصعب وصل صندوقين ، نستعمل دائرة صغيرة مرققة لتمثل هذا الاتصال .



ببضاًوى البادية أو للنهاية



مستطيل لعملية حاوية أو أى عملية غير اتخاذ القرار .



معين لاتخاذ القرار

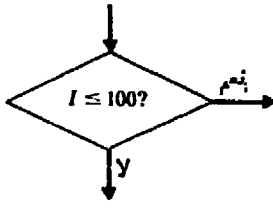


متوازي أضلاع للإدخال أو الإخراج



دائرة صغيرة للوصل

شكل ٤ - ١



يشير دائماً الصندوق ذو الشكل المدين إلى إتخاذ قرار ولذا سيخرج منه خطان أو أكثر .
 تعنون هذه الخطوط بنتائج القرار المختلفة أي بـ « نعم » أو « لا » أو بـ « حقيق » أو
 « غير حقيق » أو بـ « يجب » أو « سالب » أو « صفر » . على سبيل المثال يمكن أن يظهر
 في خريطة سير العمليات السؤال (هل $100 \geq I$ ؟) كما هو موضح على اليسار (وللتسهيل ،
 سنحذف عادة علامة الاستفهام)

عند هذه النقطة ، سنذكر مرة أخرى أن الرمز المقبول عالمياً لتخصيص قيمة لمتغير هو السهم
 ولكن ما يقابله في الفورتران هي العلامة = . وحيث أن خريطة سير العمليات مستقلة عن أي لغة ،
 فكثير من كتب الفورتران تفضل استعمال السهم في خرائط سير العمليات وستنبئ هذه السياسة
 بصفة عامة . وعلى سبيل المثال ، لإشارة إلى أن N قد تم زيادتها بـ 1 أي خصصت القيمة $N + 1$ إلى N سنستعمل عادة

$$N \leftarrow N + 1$$

بدلاً من

$$N = N + 1$$

أو أنف 1 إلى N

ونؤكد أن كل ما سبق مناقشته يستعمل بتوسع في كتب الفورتران .

٤ - ٢ الانتقال غير المشروط

يمكن إنجاز الانتقال غير المشروط بكتابة الجملة .

GO TO n

حيث n هي رقم جملة . وهذا يأمر الحاسب بأن يذهب ، بدون شرط ، إلى ذلك الجزء من البرنامج المبتدأ بالجملة
 التي تحمل الرقم n . واضح أن الجملة التي تحمل الرقم n يجب أن تكون جملة قابلة للتنفيذ . (انظر قسم ١ - ٨) .

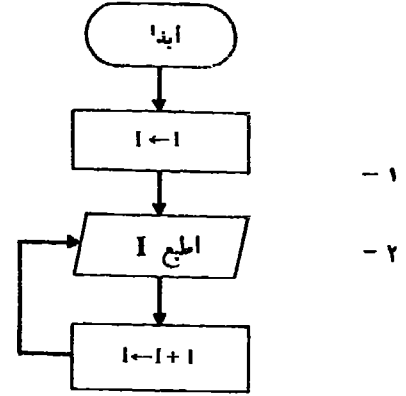
مثال ٤ - ١

(١) أكتب برنامجاً يولد ويطبغ الأرقام الصحيحة الموجبة $1, 2, 3, \dots$. نلاحظ أن أي عدد صحيح موجب يمكن الحصول
 عليه بجمع 1 على الرقم السابق له . لذلك . فهي عملية تكرار ، تم بجمع 1 على القيمة الحالية للعدد الصحيح . فإذا
 رمزنا إلى القيمة الحالية لعدد صحيح بالرمز I فإن العدد الصحيح التالي له (خليفه) هو $I + 1$ ، وحيث أن هذا العدد الصحيح
 سيستخدم مرة ثانية لحساب خليفة فسوف نخصص $I + 1$ إلى I

ليده هذه العملية ، نعطي I قيمة ابتدائية وذلك بجمع $I = 1$ ونبدأ بـ خريطة سير العمليات لهذا البرنامج وما يقابلها في
 الفورتران :

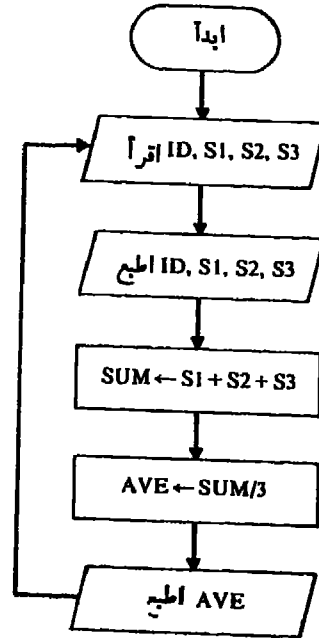
```

I = 1          I = 1
99 PRINT I    99 WRITE(6, 20) I
I = I + 1    20 FORMAT(1X, I4)
GO TO 99     I = I + 1
END          GO TO 99
            END
    
```



(ب) تحتوي كل بطاقة من مجموعة البطاقات المعطاة على أربعة أرقام تمثل رقم تحقيق الشخصية (ID) للتلميذ ، وكذا ثلاث درجات إختبار .
اكتب برنامجاً لحساب ، متوسط الاختبارات الثلاثة لكل بطاقة .

مرة ثانية ، نكرر عملية حساب المتوسط لثلاثة أرقام . لذلك نحتاج إلى انتقال غير مشروط . خريطة سير العمليات الخاصة بهذه العملية موضحة في الشكل ٤ - ٢



شكل ٤ - ٢

وفيما يلي برنامج الفورتران ، المقابل لخريطة سير العمليات :

```
C
C PROGRAM TO COMPUTE THE AVERAGE
C FOR EACH STUDENT
C
100 READ, ID, S1, S2, S3
    PRINT, ID, S1, S2, S3
    SUM = S1 + S2 + S3
    AVE = SUM/3.0
    PRINT, AVE
    GO TO 100
    END
```

وإذا استخدمنا مدخل / مخرج مصاغ يكون :

```
100 READ(5, 10) ID, S1, S2, S3
10  FORMAT(I5, F10.2, F10.2, F10.2)
    WRITE(6, 10) ID, S1, S2, S3
    SUM = S1 + S2 + S3
    AVE = SUM/3.0
    WRITE(6, 20) AVE
20  FORMAT(1X, 'THE AVERAGE IS', F8.2)
    GO TO 100
    END
```

لاحظ أن في كل من المثالين السابقين ، تتكرر العملية مراراً بدون توقف ويقال أن البرنامج كون حلقة تكرارية لانهاية . ومن الواضح أننا نحتاج لبعض أوامر لاتخاذ القرارات لإنهاء العملية كما يجب . ذلك يقودنا إلى القسم التالي ، ونشير هنا إلى أن قدرات الحاسب على اتخاذ القرار هي التي تجعل الحاسب فعال ، وفي منتهى القوة .

٤ - ٣ الانتقال المشروط

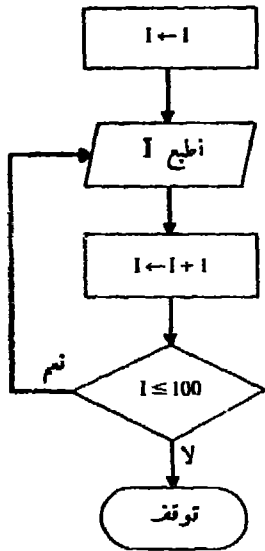
اعتبر مرة ثانية المثال ٤ - ١ (أ) لتوليد الأعداد الصحيحة . افترض أننا نريد أن نولد مائة عدد صحيح موجب فقط . أي ، افترض أننا نريد أن نهي البرنامج بعد أن نولد العدد الصحيح 100 . إحدى الطرق للتوصل إلى ذلك هو أن نسأل السؤال التالي عند النقطة A من الشكل ٤ - ٣ (أ) . أي بعد تنفيذ $I = I + 1$ نسأل :

هل القيمة الحالية لـ $I \geq 100$ ؟

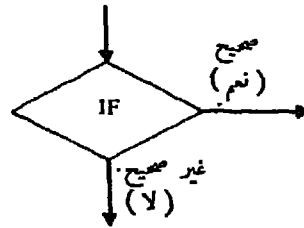
إذا كانت الإجابة نعم ، كرر العملية ، وإلا فإنه العملية . بمعنى آخر يمكن أن يعطى الأمر التالي عند النقطة A وذلك باستخدام روابط اللغة الإنجليزية المادية :

```
IF  $I \leq 100$  THEN كرر العملية
      ELSE توقف
```

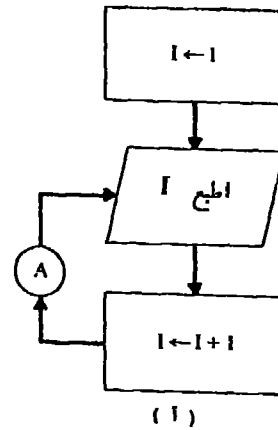
يمثل أمر IF...THEN...ELSE... هذا بصندوق على شكل معين للقرار . كما في الشكل ٤ - ٣ (ب) خريطة سير العمليات التي تولد أول مائة عدد صحيح موجب مطعاة و الشكل ٤ - ٤ .



شكل ٤ - ٤



(ب)



(١)

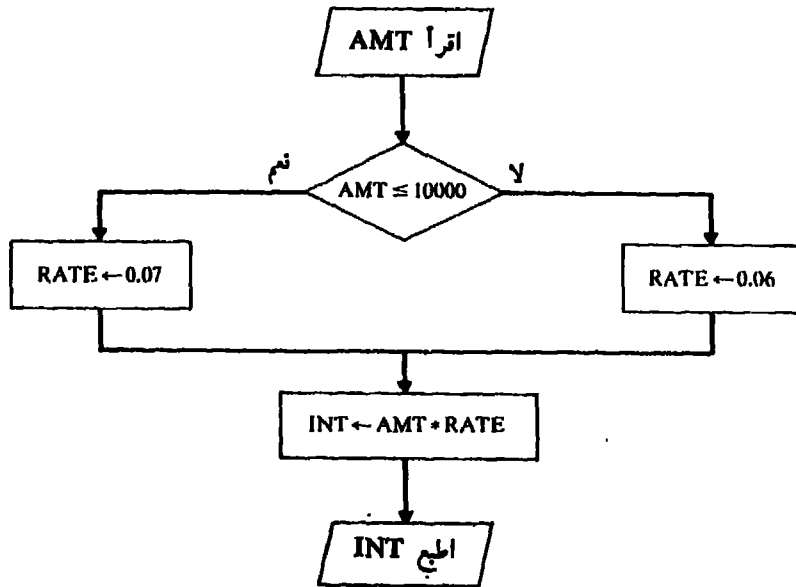
شكل ٣ - ٤

اعتبر مثالا آخر ، افرض مقدار قرض AMT مشقبا على بطاقة ، وافرض أن RATE نسبة الفائدة هي 7 في المائة إذا كانت $AMT \leq \$10,000$ ، وتكون نسبة الفائدة 6 في المائة إذا كانت AMT أكبر من $\$10,000$. ونريد أن نكتب جزء البرنامج لحساب الربح INT . نريد أن نعطي الأمر التالي بعد قراءة AMT :

```

IF AMT ≤ 10,000 THEN RATE = 0.07
ELSE RATE = 0.06
  
```

مرة أخرى ، نرى الرابط IF...THEN...ELSE... وتظهر خريطة سير العمليات لكل هذا الجزء من البرنامج في الشكل ٤ - ٥ .



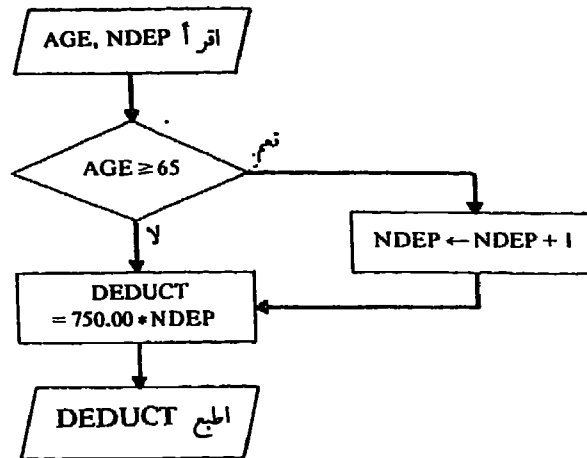
شكل ٤ - ٥

وعادة يفترض الرابط IF...THEN..ELSE.... وجود بدليين مختلفين كما هو موضح في المثالين السابقين . وعلى أى حال فتوجد حالات تتطلب استعمال الرابط الإنجليزي IF...THEN ونبين ذلك بمثال ثالث

افرض أن عمر رجل AGE وعدد الذين يعولهم NDEP بيانات مثقبة على بطاقة . ونريد أن نحسب الضرائب المخصومة على دخله DEDUCT وهو \$ 750 لكل معول وإذا كان الرجل عنده 65 سنة أو أكثر ، فيمكن أن يطالب بمخصم إضافي (نفرض أنه لا يوجد أحد من معوليه عنده ٦٥ سنة أو أكثر) وبذلك فتريد أن نعطي الحاسب الأمر التالي :

IF AGE ≥ 65 THEN NDEP = NDEP + 1

قبل حساب DEDUCT تظهر خريطة سير العمليات لمثل جزء البرنامج هذا في الشكل ٤ - ٦ وفيما يلي معنى الجملة "IF X THEN Y" حينما تتحقق X نفذ Y أولاً قبل أن تنتقل إلى الجملة التالية ، وحينما لا تتحقق X انتقل إلى الجملة التالية فوراً (أى تخطى Y) .



شكل ٤ - ٦

والفرق الدقيق بين الرابطين IF...THEN و IF..THEN..ELSE هو فرق غير مهم . ويناقش الفورتران الهيكل الذى يمكنه تنفيذ هذه الروابط في قسم ٤ - ٥ وفى الفصل الثانى عشر .

{ - ٤ } تعبيرات مترابطة

وقبل أن نقدم كيف ينفذ الفورتران الانتقال المشروط ، سنبين أولاً كيف نكتب بعض الجمل المترابطة بالفورتران .

يعطى جدول ٤ - ١ قائمة بستة ماملات ترابطية في الفورتران تمثل العلاقات الرياضية الأكثر شيوعاً بين التعبيرات الحسابية . لاحظ أن كل مكافئ بالفورتران يتكون من أربعة حروف حرفان أبجديان تسبقهما وتبهما نقطة . وجود النقط ضرورى ، والا سيفسر الحاسب الحروف الأبجدية ، ولتكن LT على أنها متبعر فضلاً عن كونها مامل ترابط .

جدول ٤ - ١

العلاقة	فورتران
.LT.	أقل من
.LE.	أقل من أو يساوي
.EQ.	يساوي
.NE.	لا يساوي
.GT.	أكبر من
.GE.	أكبر من أو يساوي

يتكون التعبير المترابط من تعبيرين حايين يصل بينهما معامل ترابط واحد . والتعابير المترابطة هي أبسط أشكال التعابير المنطقية (سنناقش تعابير منطقية تحتوي على أدوات منطقية أكثر تعقيداً في الفصل التاسع) . وعند وقت التنفيذ يمثل كل تعبير منطقي ، وبالتالي كل تعبير مترابط شرطاً إما أن يتحقق أو لا يتحقق .

مثال ٤ - ٢

وفيما يلي تعبيرات رياضية والتعابير المترابطة المكافئة لها في الفورتران :

$I < J$	I.LT.J
$(A + 3) > B$	(A + 3.0).GT.B
$5I = 2J$	5*I.EQ.2*J
$B \geq A^2$	B.GE.A**2
$(I + 3) \neq J$	(I + 3).NE.J
$(A + B^3) \leq 50$	(A + B**3).LE.50.0

افرض أن I و J و A و B تحتوي 2 و 5 و 3.0 و 4.0 على الترتيب . ومن ثم فالتعابير الثلاثة المترابطة الأولى تتحقق والثلاثة الأخرى لا تتحقق .

يجب أن نذكر هذين التعليقين :

- ١ - تجنب مقارنة تعبير صحيح بتعبير حقيق لأن القيم الصحيحة تختلف عن القيم الحقيقية في التمثيل الداخلي للذاكرة .
- ٢ - يجب أن نتوخى الحرص عند استعمال .EQ. و .NE. بين الأرقام الحقيقية . بسبب الخطأ النظري في الحسابات ، قد لا تتساوى إطلافاً قيمتان حقيقيتان داخل الحاسب مع أن المفروض نظرياً أنهما متساويان . ولذا فبدلاً من استخدام A.EQ.B فغالباً نختبر اختلافهما بشئ من التفاوت مثل

$$\text{ABS}(A - B).\text{LT}.0.0001$$

للتعويض عن أخطاء التقريب الممكنة .

٤ - ٥ جملة IF المنطقية

يمكن أن تم الانتقالات المشروطة بعدة طرق في الفورتران وإحدى هذه الطرق هي استعمال جملة IF المنطقية وشكلها العام .

IF(logexp) Statement A

وبالتحديد ، تبدأ جملة IF المنطقية بكلمة IF ويتمها تعبير منطقي log exp بين قوسين (تذكر أن التعبير المنطقي log exp هو تعبير له القيمة FALSE أو TRUE انظر أيضاً الفصل التاسع) وبعد ذلك تتبع الاقواس جملة قابلة للتنفيذ (Statement A) ولذلك فالجملة A يجب ألا تكون جملة IF أخرى أو جملة DO (سنناقش ذلك في الفصل الخامس) .

فيما يلي أمثلة لجمل IF المنطقية :

```
IF(A.LE.B) GO TO 50
IF(L.GE.75) N = N + 1
IF(J.NE.K) WRITE(6, 30) X, Y.
```

وبطريقة رسمية ، فجملة IF المنطقية

IF(logexp) Statement A

لها المعنى التالي : إذا تحقق التعبير المنطقي (Log exp) أى (TRUE) نفذت الجملة A وإلا تحطى الجملة A

وبأكثر دقة ، نفترض جزء من برنامج فورتران يحتوي

**IF(logexp) Statement A
Statement B**

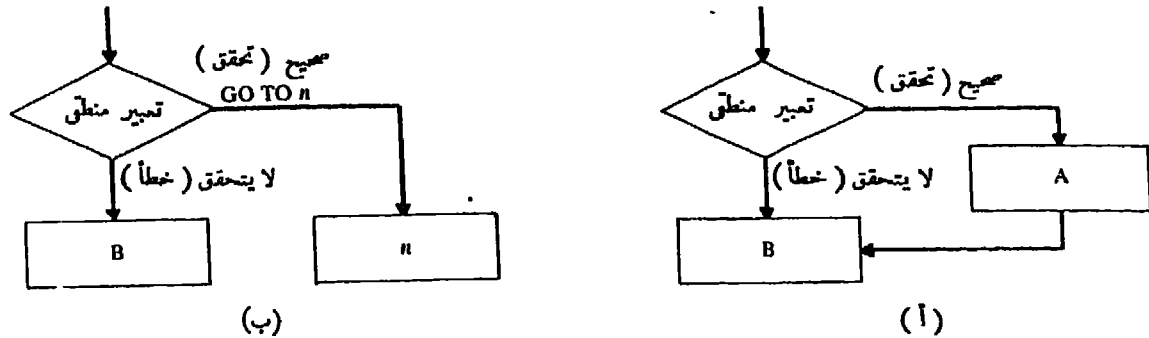
حيث جملة B هي أول جملة قابلة للتنفيذ تتبع جملة IF . وبذلك يمكن أن يحدث الموقفان التاليان :

١ - جملة A ليست جملة GO TO

٢ - جملة A هي جملة GO TO

الحالة ١

إذا كانت جملة A ليست جملة GO TO فجزء برنامج الفورتران السابق يعنى الآتي . عندما يتحقق التعبير المنطقي أى أن (Log exp) تكون TRUE نفذ الجملة A أولاً ثم أكمل إلى الجملة التالية ، أى ، جملة B ؛ وعندما لا يتحقق التعبير المنطقي أى أن (Log exp) تكون FALSE ، أكمل فقط إلى الجملة التالية أى جملة B (متخطياً الجملة A) ونوضح هذا الموقف في الشكل ٤ - ٧ (أ) . لاحظ أن جملة B تكون دائماً جملة منفذة . وتكافئ هذه الخريطة الاستعمال العادي للرابط الإنجليزي IF...THEN... الذي تمت مناقشته في قسم ٤ - ٣ .



شكل ٤ - ٧

الحالة ٢

افترض أن جملة A هي جملة GO TO أى أن جزء برنامج الفورتران السابق هو

**IF(logexp) GO TO n
Statement B**

وهذا يعنى الآت : إذا تحقق التعبير المنطوق أى أن $(Log\ exp)$ تكون TRUE ينفذ جزء البرنامج المبتدأ بالجملة رقم 77 . أما إذا لم يتحقق التعبير المنطوق وأى أن $(Log\ exp)$ تكون FALSE ينفذ جزء البرنامج مبتدأ بالجملة B . تظهر خريطة سير العمليات فى شكل ٤ - ٧ (ب) . وهى توضح أن بديلين مختلفين يمكن حدوثهما بعد القرار ، وهذا يكافئ الاستخدام العادى للرباط IF... THEN...ELSE... وبذلك يمكن أن يستعمل هذا الشكل من جملة IF منطقية كلما ظهر IF...THEN...ELSE... فى الخوارزم (النظام الحسابى) . (انظر أيضاً الفصل الثانى عشر) .

والآن ادرس خريطة سير العمليات فى شكل ٤ - ٥ التى تحسب الربح INT على قرض AMT وتذكر أن خريطة سير العمليات تم رسمها كنتيجة للأمر .

```
IF AMT ≤ 10,000 THEN RATE = 0.07
ELSE RATE = 0.06
```

فقرض أن جزء برنامج الفورتران

```
IF(AMT.LE.10000) RATE = 0.07
RATE = 0.06
```

قد استخدم لتنفيذ الأمر السابق . فتمياً لشكل ٤ - ٧ (أ) والمناقشة التى تمت على الصفحة السابقة فإن $RATE = 0.06$ سوف تنفذ دائماً . ولكن ليس هذا هو ما نريد لذا لا يمكن استخدام هذا الجزء من برنامج الفورتران .

من نتخطى $RATE = 0.06$ حين يكون $AMT ≤ 10,000$ يجب أن نستخدم جملة GO TO فى جملة IF التى

كافى برنامج الفورتران التالى :

```
REAL INT
READ(5, 10) AMT
10 FORMAT(F15.2)
IF(AMT.LE.10000.) GO TO 100
RATE = 0.06
GO TO 200
100 RATE = 0.07
200 INT = AMT*RATE
WRITE(6, 20) INT
20 FORMAT(1X, 'THE INTEREST IS', 2X, F10.2)
STOP
END
```

مثال ٤ - ٣

(أ) فترجم خريطة سير العمليات في شكل ٤ - ٤ إلى فورتران وهي تليج أول 100 عدد صحيح موجب . لاحظ أن هناك طريقتين متفرقتين بعد القرار ولذلك فإننا نتوقع أن نستخدم الشكل « *IF Log exp GO TO* » من جملة *IF* المنطقية وفيما يلي هذه الترجمة .

```
I = 1
99 WRITE(6, 10) I
10 FORMAT(1X, I5)
   I = I + 1
   IF(I.LE.100) GO TO 99
   STOP
   END
```

الخطتان

```
IF(I.LE.100) GO TO 99
STOP
```

هما ترجمة الأمر التالي إلى فورتران :

```
IF I ≤ 100 THEN GO TO 99 to repeat the process
ELSE STOP
```

(ب) وثيما يلي ترجمة خريطة سير العمليات ، التي تحسب الضرائب المخصومة على دخل شخص ، *DEDUCT* إلى فورتران :

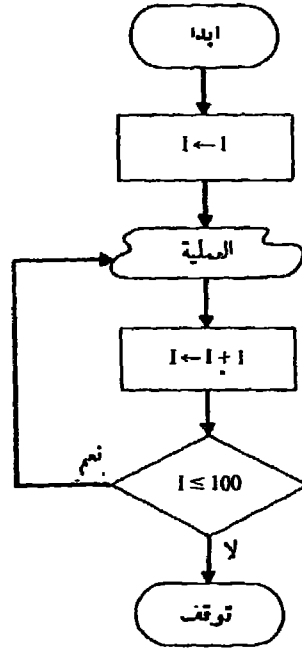
```
INTEGER AGE
READ(5, 10) AGE, NDEP
10 FORMAT(2I15)
   IF(AGE.GE.65) NDEP = NDEP + 1
   DEDUCT = 750.00*FLOAT(N)
   WRITE(6, 20) DEDUCT
20 FORMAT(1X, F12.2)
   STOP
   END
```

لاحظ التماثل بين الشكل ٤ - ٧ (ب) وجزء من الشكل ٤ - ٦ . لاحظ أيضاً أننا استعملنا الدالة المكتيبة *FLOAT* لتحويل العدد الصحيح *N* إلى عدد حقيقي حتى نتجنب حساب النمط المختلط .

٤ - ٦ التحكم في الحلقة التكرارية

افترض أننا نريد تكرار عملية (أي مجموعة من الأوامر) وليكن 100 مرة . يمكننا عمل ذلك باستخدام عداد *I* يمد عدد مرات تكرار العملية . أي أننا نخصص 1 للعداد *I* في البداية وبعد ذلك ، في كل مرة ننفذ فيها العملية ، نزيد قيمة العداد *I* بمقدار

واحد (1) ونهى تنفيذ العملية إذا تعدت قيمة المداد [الرقم 100 ويوضح شكل ٤ - ٨ خريطة سير العمليات التي نفذ منه المهمة .



شكل ٤ - ٨

فلاحظ في التو التماثل بين شكل ٤ - ٤ وشكل ٤ - ٨ . وبمعنى آخر فيمكن أيضاً أن نستخدم ميكانيكية التحكم التي استخدمناها لتوليد أول 100 عدد صحيح موجب كعملية للمد والتحكم في الخروج من حلقة تكرارية وذلك بتكرارها 100 مرة

مثال ٤ - ٤

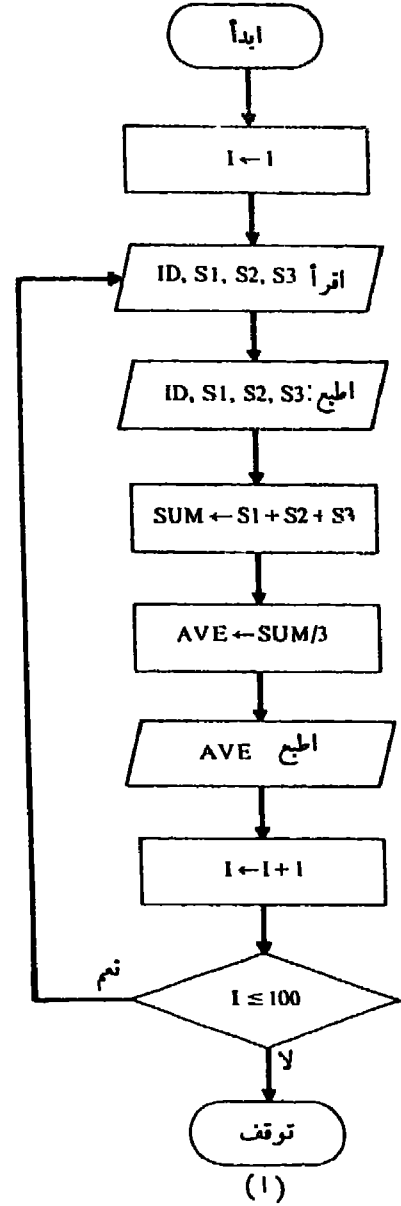
استرجع مثال ٤ - ١ (ب) حيث أوجدنا متوسط ثلاث درجات لكل طالب . اكتب البرنامج إذا كان هناك 100 بطاقة في مجموعة بطاقات البيانات .

نحن ندرج ببساطة خريطة سير العمليات لمثال ٤ - ١ (ب) في المكان المناسب من شكل ٤ - ٨ لنحصل على شكل ٤ - ٩ (أ) . ترجمة الفورتران لبرنامجنا موضحة في شكل ٤ - ٩ (ب) لاحظ أن ترجمة خريطة سير العمليات إلى فورتران عملية مباشرة عموماً .

```

C
C
C
C
C
C
C
C
C
C
PROGRAM TO COMPUTE THE AVERAGE OF THREE TESTS
C
C
C
C
C
INITIALIZE THE COUNTER I
C
I = 1
C
C
C
C
PROCESS TO BE REPEATED
C
100 READ(5, 10) ID, S1, S2, S3
10  FORMAT(14, 3F6.2)
   WRITE(6, 10) ID, S1, S2, S3
   SUM = S1 + S2 + S3
   AVE = SUM/3.0
   WRITE(6, 20) AVE
20  FORMAT(20X, 'THE AVERAGE IS', F6.2)
C
C
C
C
INCREMENT THE COUNTER I
C
I = I + 1
IF(I.LE.100) GO TO 100
STOP
END
    
```

(ب)



(أ)

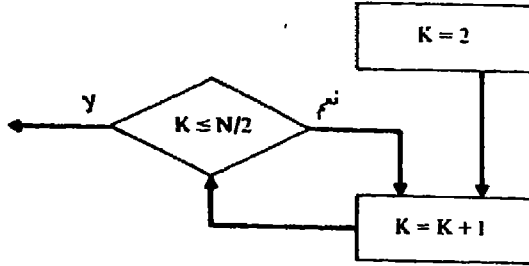
شكل ٤ - ٩

مثال ٤ - ٥

افترض أننا نريد كتابة برنامج لقراءة عدد صحيح $N > 2$ ونحدد ما إذا كانت N عدد أولي أم لا .

تذكر أن N تكون عدداً أولياً إذا كانت لا تقبل القسمة إلا على نفسها وعلى الواحد الصحيح على سبيل المثال والأعداد 2 و 3 و 5 و 13 و 29 و 67 أعداد أولية . ولكن 35 ليس عدداً أولياً حيث أنه يقبل القسمة على 5 ، 7 . ومن الواضح أنه إذا كانت N ليست عدد أولياً فإن N تقبل القسمة على أحد هذه الأعداد الصحيحة 2 و 3 و ... $N/2$ (قارن المسألة ٤ - ٤٨)

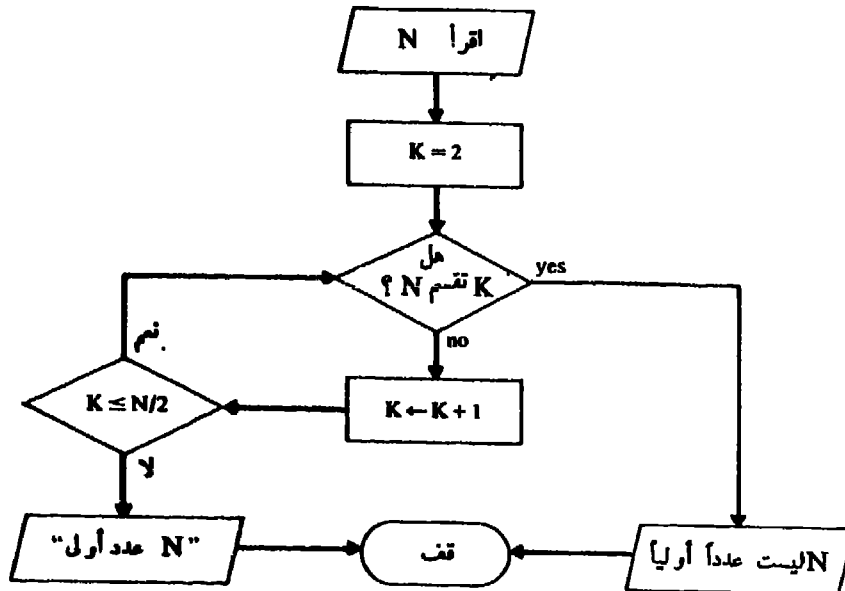
١ - ولد الأعداد الصحيحة 2 و 3 و... و $N/2$ ويمكن عمل هذا كما هو مبين في شكل ٤ - ١٠ .



شكل ٤ - ١٠

٢ - اختبر إذا كانت N تقبل القسمة على كل عدد من الأعداد الصحيحة السابقة . إذا حدث ذلك بواسطة أي منها فإن N ليست عدداً أولياً ، وإلا فتكون N عدداً أولياً :

تظهر خريطة سير العمليات للنظام الحسابي (الخوارزم) في شكل ٤ - ١١ .



شكل ٤ - ١١

قبل أن نترجم خريطة سير العمليات إلى فورتران ، يجب أن نكون قادرين على ترجمة الصئوق : **Does k Divide N?** . ومع ذلك فنحن نعرف من مثال ٢ - ٩ أن N تقبل القسمة على K إذا كانت $(N/K) \cdot K$ لها نفس قيمة N . وبذلك ، يكون لدينا انبرناسج :

```

READ(5, 20) N
20  FORMAT(I5)
   K = 2
30  IF(N/K*K.EQ.N) GO TO 70
   K = K + 1
   IF(K.LE.N/2) GO TO 30
   WRITE(6, 40) N
40  FORMAT(1X, I5, 1X, 'IS A PRIME')
   STOP
70  WRITE(6, 50) N
50  FORMAT(1X, I5, 1X, 'IS NOT A PRIME')
   STOP
END

```

٤ - ٧ جملة IF الحسابية

يوجد نوع آخر من جمل IF ضمن مجموعة الفورتران وتسمى جملة IF الحسابية . وعند مقارنتها بجملة IF المنطقية ، التي تختبر الحالة المنطقية لتعبير منطقي : نجد أن جملة IF الحسابية تختبر إشارة تعبير حسابي . ويمكن أن ينتقل التحكم بعد ذلك إلى أماكن متعددة تبعاً لكونها موجبة أو صفر أو سالبة .

وفيما يلي شكل جملة IF الحسابية :

IF(expr) l, m, n

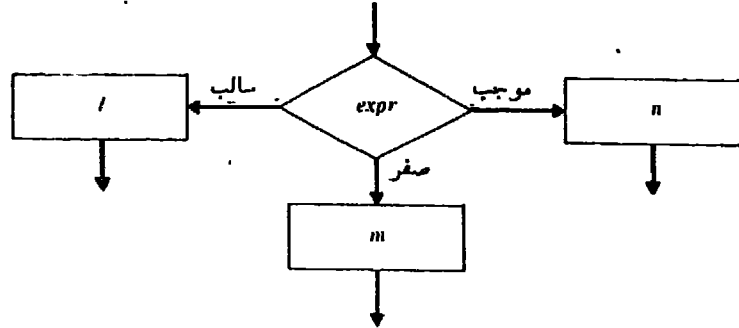
حيث *expr* هو تعبير حسابي و *l, m, n* أرقام جمل وتقول الجملة :

إذا كانت قيمة *expr* سالبة (< 0) اذهب إلى الجملة رقم *l*

وإذا كانت قيمة *expr* صفراً ($= 0$) اذهب إلى الجملة رقم *m*

إذا كانت قيمة *expr* موجبة (> 0) اذهب إلى الجملة رقم *n*

يجب أن تكون الجمل التي تحمل أرقام *l, m, n* جملاً قابلة للتنفيذ ، وليس من المطلوب أن تكون جميعها مختلفة : بمعنى أن أي اثنين منها أو جميعها يمكن أن تكون نفس الشيء . وخريطة سير العمليات المكافئة لجملة IF الحسابية موضحة في شكل



شكل ٤ - ١٢

يمكن لأي جملة IF حسابية أن تنفذ بتكافؤ إذا استخدمنا واحدة جملة IF المنطقية مع جملة GO TO أو أكثر فعل سيبل المثال :

IF(X - 2.0) 10, 20, 30 and IF(X - 2.0) 10, 10, 20 (٤)

يكافئان على الترتيب :

IF(X.LT.2.0) GO TO 10 IF(X.LE.2.0) GO TO 10
IF(X.EQ.2.0) GO TO 20 and GO TO 20
GO TO 30

وبالعكس ، فيمكن لأي جملة IF منطقية أن تنفذ بتكافؤ باستعمال جملة IF الحسابية (انظر مسألة ٤ - ٥) يحذر ملاحظة أن بعض المسائل المحددة تناسب بصورة طبيعية جملة IF الحسابية كما نرى في المثال التالي :

مثال ٤ - ٦

(أ) افترض أن قسط التأمين الصحي الذي يخصم من مرتب موظف يحسب تبعاً للخطوة التالية :

إذا كان أعزب	9.75	} = قسط
إذا كان متزوجاً ولا يعمل	16.25	
إذا كان متزوجاً ويعمل	24.50	

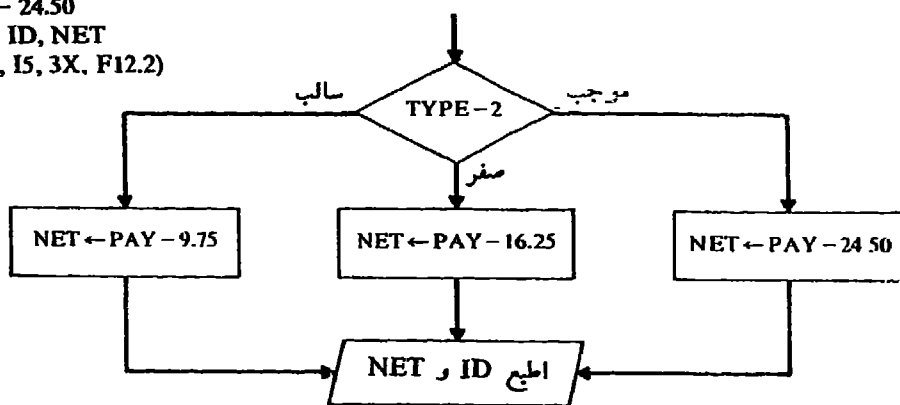
افترض أن TYPE هي المتغير الصحيح الذي يشير إلى الحالة الاجتماعية بأكواد كما يلي :

1 أعزب ، 2 متزوج ولا يعمل ، 3 متزوج ويعمل . يعطى شكل ٤ - ١٣ خريطة سير العمليات بجزء من برنامج يخصم القسط من مرتب الموظف إذا افترضنا معرفة PAY و TYPE و ID للموظف . لاحظ أن 2 - TYPE تكون سالبة ، أو صفر ، أو موجبة تبعاً لقيمة TYPE إذا كانت 1 أو 2 أو 3 . فيما يلي ترجمة شكل ٤ - ١٣ إلى فورتران :

```

REAL NET
IF(TYPE - 2) 10, 20, 30
10 NET = PAY - 9.75
GO TO 75
20 NET = PAY - 16.25
GO TO 75
30 NET = PAY - 24.50
75 WRITE(6, 40) ID, NET
40 FORMAT(1X, I5, 3X, F12.2)

```



شكل ٤ - ١٣ (ب) ادرس المعادلة التربيعية :

$$ax^2 + bx + c = 0$$

حيث a و b و c أرقام حقيقية و $a \neq 0$ ستكون هناك جذور حقيقية فقط عندما يكون المميز $D = b^2 - 4ac$ غير سالب وتمطلي الجذور بواسطة :

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

لاحظ أنه إذا كان المميز $D = 0$ فإننا نحصل على الجذر المتعدد $-b/2a$ — اكتب برنامج فورتران يحسب الجذور الحقيقية لمجموعة معطاه من المعاملات A, B, C

```

C
C COMPUTE THE REAL ROOTS OF A QUADRATIC EQUATION WHERE A
C IS NOT ZERO
C
C READ(5, 11) A, B, C
11 FORMAT(3F10.2)
WRITE(6, 12) A, B, C
12 FORMAT('0', 'THE COEFFICIENTS ARE', 3(2X, F10.2))
D = B**2 - 4.0*A*C
C
C TEST THE DISCRIMINANT
C
C IF(D) 22, 33, 44
22 WRITE(6, 13)
13 FORMAT(1X, 'THERE ARE NO REAL ROOTS')
GO TO 10
33 ROOT = -B/(2.0*A)
WRITE(6, 14) ROOT, ROOT
14 FORMAT(1X, 'THERE ARE TWO IDENTICAL ROOTS', 2(3X, F10.2))
GO TO 10
44 ROOT1 = (-B + SQRT(D))/(2.0*A)
ROOT2 = (-B - SQRT(D))/(2.0*A)
WRITE(6, 15) ROOT1, ROOT2
15 FORMAT(1X, 'THERE ARE TWO DISTINCT ROOTS', 2(3X, F10.2))
10 STOP
END

```


٤ - ٨ جملة GO TO المحسوبة

جملة (٤) GO TO المحسوبة هي جملة أخرى من جمل التحكم المشروط . في الحقيقة ، فهذه الجملة تسمح بأي عدد من اشتراطات التنقل . ويعتمد قرار التنقل المعين على قيمة متغير صحيح يظهر في الجملة .

وفيما يلي الشكل العام لجملة GO TO المحسوبة :

GO TO (n_1, n_2, \dots, n_k), J

حيث n_1, n_2, \dots, n_k ثوابت صحيحة بدون إشارة وتمثل أرقام جمل و J هي متغير صحيح بدون دليل . لاحظ أن أرقام سطر تفصل عن بعضها بواسطة فاصلات ونحاط بأقواس ، والأقواس متبوعة بفصلة أخرى ثم بعد ذلك اسم المتغير J .

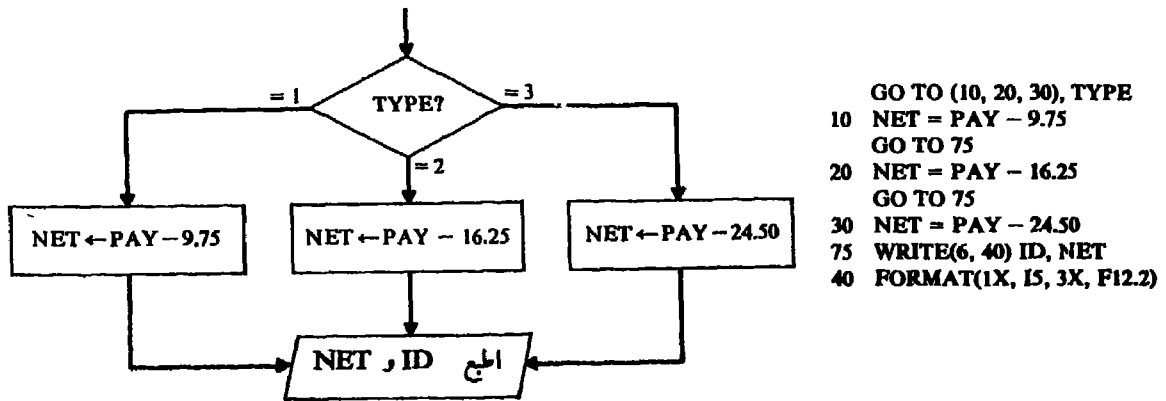
وجملة GO TO المحسوبة السابقة لها المعنى التالي : عند وقت التنفيذ « تختبر قيمة المتغير الصحيح J أولاً . إذا كانت قيمة J هي 1 ، انتقل التحكم إلى الجملة التي تحمل الرقم n_1 (أي ، عندئذ تنفذ الجملة رقم n_1) ، وإذا كانت قيمة J هي 2 عندئذ تنفذ الجملة التي تحمل الرقم n_2 وهكذا ، وأنه لغاية في الأهمية أن تقع قيمة J بين 1 و k أثناء وقت التنفيذ وإلا لا نستطيع أن نتنبأ بالنتيجة . في الحقيقة ، أن بعض المترجمات لا تعطى حتى رسالة خطأ .

افترض على سبيل المثال ، أنه تم تنفيذ الجملة التالية :

GO TO (10, 15, 70, 22, 15), NEW

يفتدى التحكم إلى الجملة التي تحمل الرقم 10 أو 15 أو 70 أو 22 أو 15 تبعاً للقيمة التي تأخذها NEW التي تأخذها 1 أو 2 أو 3 أو 4 أو 5 على الترتيب . لاحظ أن أرقام الجمل ليس من المطلوب أن تكون مختلفة . ولكن نؤكد أنه عند وقت التنفيذ يجب أن تكون NEW موجبة ولا تتعدى 5 .

ادرس مثال ٤ - ٤ حيث يخضع قسط التأمين الصحي تبعاً لكون الموظف أعزب (TYPE 1) أو متزوجاً ولا يعمل (TYPE 2) أو متزوجاً ويعمل (TYPE 3) يظهر في شكل ٤ - ٤ خريطة سير العمليات وجزءاً من برنامج الفورتراڤ باستخدام جملة GO TO المحسوبة .



شكل ٤ - ٤

يمكن أن نرى بسهولة أن جملة GO TO المحسوبة لا تعطينا أى تسهيلات حماية إضافية . أى أننا يمكن أن ننظم سلسلة من جمل IF المنطقية لإنجاز نفس نتيجة جملة واحدة من جمل GO TO المحسوبة . على سبيل المثال :

```
IF(MM.EQ.1) GO TO 99
IF(MM.EQ.2) GO TO 88
IF(MM.EQ.3) GO TO 101
IF(MM.EQ.4) GO TO 23
```

تكاليف

GO TO (99, 88, 101, 23), MM

مع ذلك فمجموعة جمل IF المنطقية أقل كفاءة من جملة واحدة من جمل GO TO المحسوبة ، والتي تتطلب خطوة واحدة فقط لنقل التحكم إلى المكان المطلوب . وكما نرى أنها تحتاج إلى كتابة أقل .

ملاحظة : هناك نوع آخر من جمل التحكم المشروط والتي تسمح بالتفرع المتعدد ، وهى جملة GO TO المخصصة . وستناقش هذه الجملة مع جملة ASSIGN المصاحبة في الفصل الحادى عشر .

٤ - ٩ الخوارزميات (نظم الحساب)

سنناقش برنامجين في هذا القسم . والفرض هو توضيح المراحل المختلفة المتعلقة بكتابة البرنامج بدءاً من وضع النظام الحسابى (الخوارزم) ثم تهيئته ، إلى تنفيذه ، ولن تم ترجمة خرائط سير العمليات النهائية هنا إلى فورتران ، بل ستترك كتمرينات (مسائل تكميلية) للقارىء .

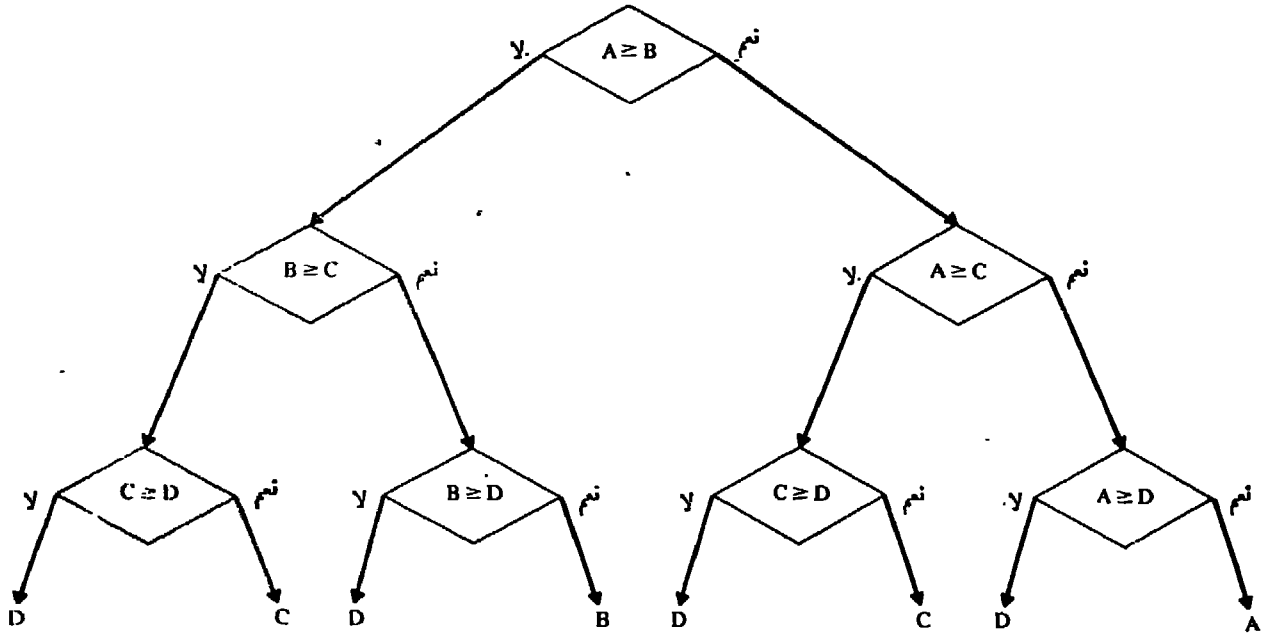
افترض أنه تم إعطاء مجموعة من أربع بطاقات مشقبة على كل بطاقة عدد . والمطلوب إيجاد أكبر عدد من هذه الأعداد . ولما كان هناك أربعة أعداد فقط . دعنا نقرأها داخل الذاكرة ونطلق عليها A و B و C و D . سيمطينا النظام الحسابى (الخوارزم) التالي أكبر عدد بفرض أن A و B و C و D في الذاكرة .

١- قارن A و B وأوجد أيهما أكبر .

٢- قارن C مع الناتج من 1 وأوجد أيهما أكبر .

٣- قارن D مع الناتج من 2 وأوجد أيهما أكبر .

وخريطة سير العمليات في شكل ٤ - ١٥ توضح أحد الأساليب المباشرة والبديهة للنظام الحسابى (الخوارزم) السابق .



شكل ٤ - ١٥

لاحظ أن هناك سبعة ($7 = 1 + 2^1 + 2^2$) صناديق اتخاذ قرار في شكل ٤ - ١٥ وفي الحقيقة ، العدد الكلي لصناديق اتخاذ القرار في مثل هذا النظام الحسابي (الحوارزم) تنمو أسياً مع عدد العناصر ، بالتحديد لو كان هناك خمسة أرقام ، سيكون هناك عشرة صندوق لاتخاذ قرار ، ولو كان هناك ستة أرقام ، سيكون هناك ($31 = 15 + 2^4$) إحدى وثلاثون صندوق لاتخاذ قرار ، وهكذا من الواضح أن هذا النظام الحسابي (الحوارزم) ليس مقنماً عندما يكون عدد العناصر كبيراً .

نهدف نظامنا الحسابي (الحوارزم) قليلاً بجعل LAR تشير إلى القيمة الكبرى ونستعمل الخطوات الأربع التالية :

(٠) خصص A إلى LAR أي ضع $LAR = A$

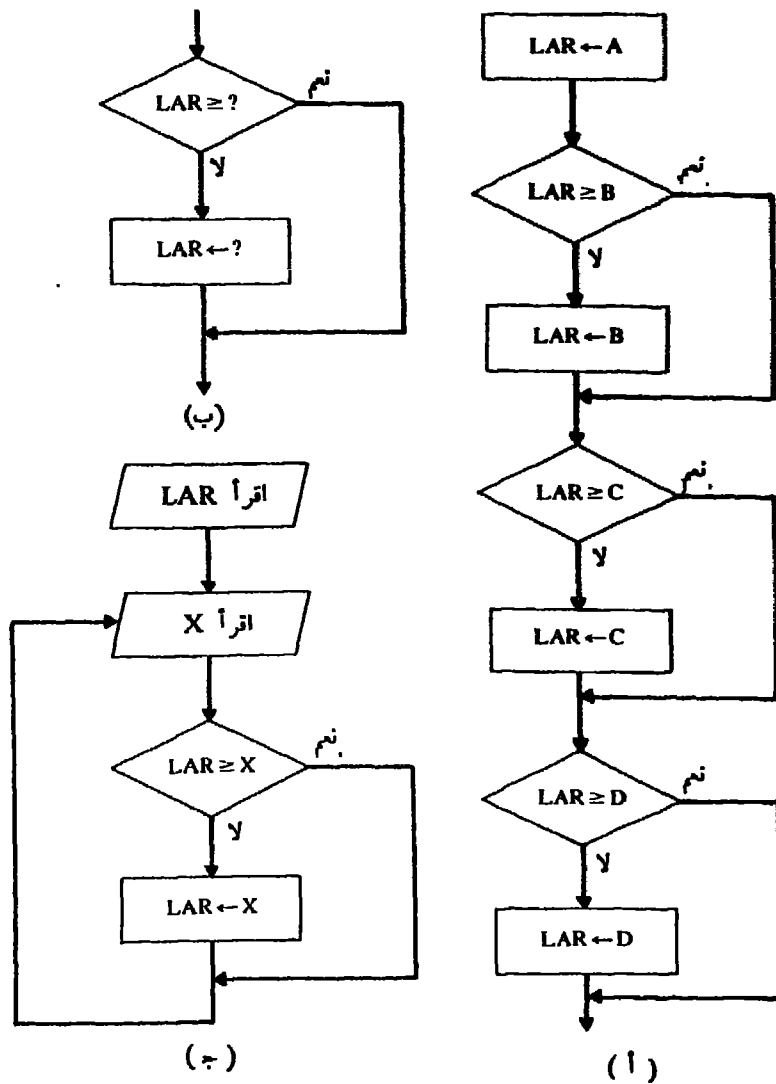
(١) قارن LAR و B واحتفظ بالأكبر في LAR

(٢) قارن LAR و C واحتفظ بالأكبر في LAR

(٣) قارن LAR و D واحتفظ بالأكبر في LAR

خريطة سير العمليات لهذا النظام الحسابي (الحوارزم) معطاه في شكل ٤ - ١٦ (أ) لاحظ أن عدد صناديق اتخاذ قرار ثلاثة ، وتنمو خطياً مع عدد العناصر . أي ، بالنسبة للأرقام الخمسة ، يكون هناك أربعة صناديق اتخاذ قرار ، ولعدد n من الأرقام ، سيكون هناك $I - n$ صناديق اتخاذ قرار . واضح ، أن هذا النظام الحسابي (الحوارزم) أكثر تقدماً من النظام الحسابي (الحوارزم) السابق .

من الأشياء الهامة التي تلفت النظر في خريطة سير العمليات بالشكل ٤ - ١٦ (أ) هو تكرار مجموعة معينة من الصناديق وسير العمليات مرة بعد أخرى . وهذه المجموعة معينة في شكل ٤ - ١٦ (ب) حيث ؟ تمثل عنصراً اختيارياً (بخالف الأول) ، وبالتالي لو استعملنا أن نضمن أن قيمة ؟ تتغير بطريقة صحيحة ، سيكون حينئذ أن نجيب الصناديق في حلقة تكرارية . ويتم هذا التغيير بقراءة القيم واحدة تلو الأخرى باستعمال نفس الاسم (مكان للتخزين) في كل مرة والحلقة التكرارية اللانهائية الناتجة عن ذلك مبينة في شكل ٤ - ١٦ (ج) .



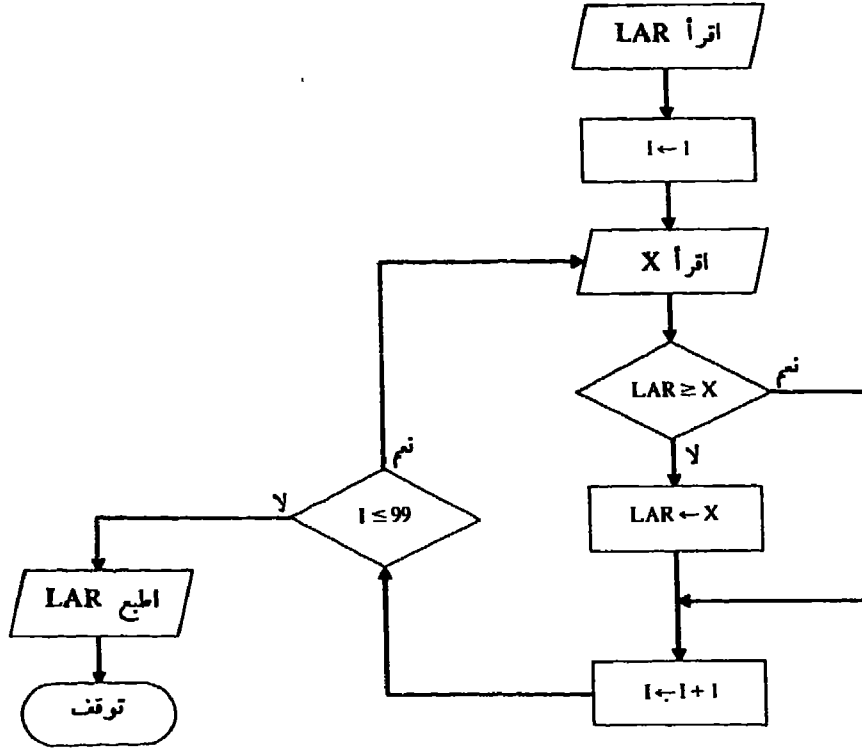
شكل ٤ - ١٦

والآن افترض أن المجموعة المغطاة 100 بطاقة . في هذه الحالة تظهر ، خريطة سير العمليات لإيجاد الرقم الأكبر كما في شكل ٤ - ١٧ (ولما كان قد خصص الرقم الأول لـ LAR في البداية فإن العملية تتكرر $100 - 1 = 99$ مرة فقط) ويظهر برنامج الفورتران لمسألة أعم من ذلك قليلا في مسألة ٤ - ١٣ .

لندرس الآن مثالا آخر . افترض مجموعة من أربع بطاقات ومثقب على كل بطاقة عدد ما . المطلوب إيجاد المتوسط (المتوسط الحسابي) لهذه الأعداد . والطريقة المباشرة هي أن تقرأ هذه الأعداد داخل أماكن تخزين ، وتكن A و B و C و D ثم تقسم مجموعها على 4 وتظهر خريطة سير العمليات في شكل ٤ - ١٨ (أ)

والآن نفرض أن مجموعة البطاقات هي 25 وليست 4 بطاقات . فباستخدام النهج السابق ، سيتطلب ذلك قراءة 25 متغيراً مختلفاً ، ليكن A_1, A_2, \dots, A_{25} (الأعداد 1 ، 2 ، ... ، 25 في أسماء المتغيرات لا تلعب أي دور سوى كونها رموزاً مختلفة) . ويمثل الشكل ٤ - ١٨ (ب) برنامج الفورتران الذي سيحسب متوسط 25 عدداً . لاحظ أن كلا من جملة READ والجملة التي تحسب SUM طويلة جداً ومن السهل حدوث أخطاء بها . كذلك يلاحظ أنه إذا كان هناك مائة رقم فيستعمل تماماً

التحكم في الحل بهذا الأسلوب . وعلاوة على ذلك ، سنضطر أيضاً لإعادة كتابة البرنامج . ويتضح لنا بالتالي أنه من الأفضل استخدام نظام حسابي (خوارزم) معين بحيث يمكن استخدام نفس البرنامج (مع احتمال تغيير طفيف) لحساب مجموع 4 أعداد أو 25 أو 100 عدد . وسنناقش هذا فيما يلي .

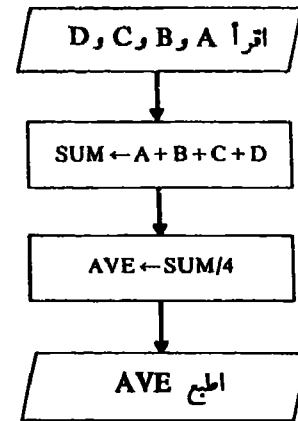


شكل ٤ - ١٧

```

1  READ(5, 10) A1, A2, A3, A4, A5, A6, A7, A8, A9, A10,
1  A11, A12, A13, A14, A15, A16, A17, A18, A19, A20,
1  A21, A22, A23, A24, A25
10 FORMAT(F10.1)
SUM = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10
1  + A11 + A12 + A13 + A14 + A15 + A16 + A17 + A18
1  + A19 + A20 + A21 + A22 + A23 + A24 + A25
AVE = SUM/25.0
WRITE(6, 20) AVE
20 FORMAT(1X, 'THE CLASS AVERAGE IS', 1X, F4.1)
STOP
END
  
```

(ب)



(أ)

شكل ٤ - ١٨

لحساب $A_1 + A_2 + A_3 + A_4 + \dots$ يمكن أن نعتبرها

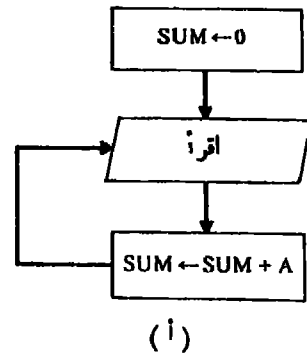
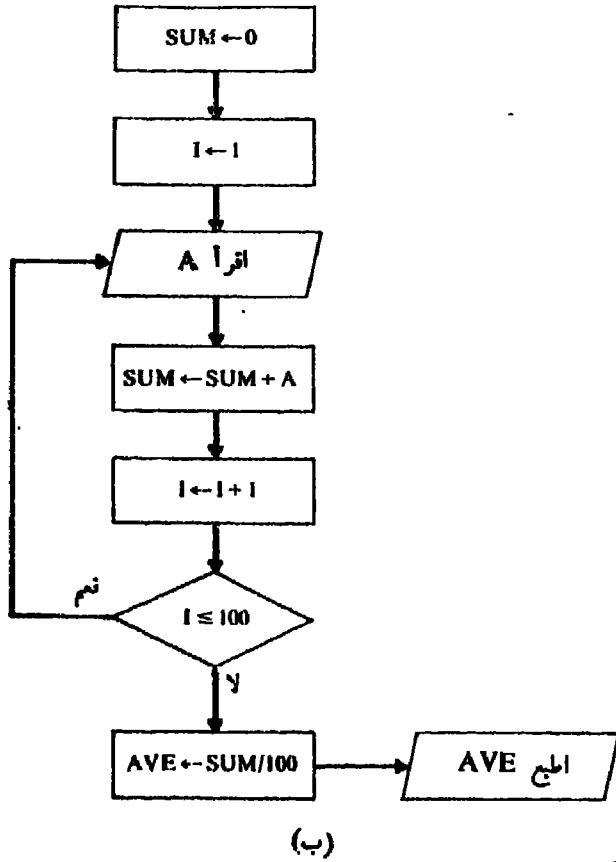
$$(((A_1) + A_2) + A_3) + A_4) + \dots$$

أى يمكن أن نبدأ بـ $SUM = 0$ ثم يتم جمع هذه الأعداد واحدة تلو الأخرى إلى محتويات SUM . في الواقع ، فإننا يمكن أن نجمعها واحدة تلو الأخرى من البداية ، أى عند قراءتها وتخزينها داخل الذاكرة ، كما هو موضح في الحلقة التكرارية اللانهائية من الشكل ٤ - ١٩ (أ). والآن إذا كان هناك مجموعة من 100 بطاقة ومثقب في كل منها عدد واحد ، فيمكن أن يجهز عدد I لعدد عدد المرات التي تنفذ فيها الحلقة التكرارية. وبذا يتم إيقاف العملية بعد تنفيذ جميع المائة عدد. وهذا يعطى خريطة سير العمليات التي في الشكل ٤ - ١٩ (ب) وفيما يلي ما يقابله بالפורتران.

```

SUM = 0.0
I = 1
50 READ(5, 10) A
10 FORMAT(F10.2)
SUM = SUM + A
I = I + 1
IF(I.LE.100) GO TO 50
AVE = SUM/100.0
WRITE(6, 20) AVE
20 FORMAT(1X, 'THE AVERAGE IS', 1X, F4.1)
STOP
END

```

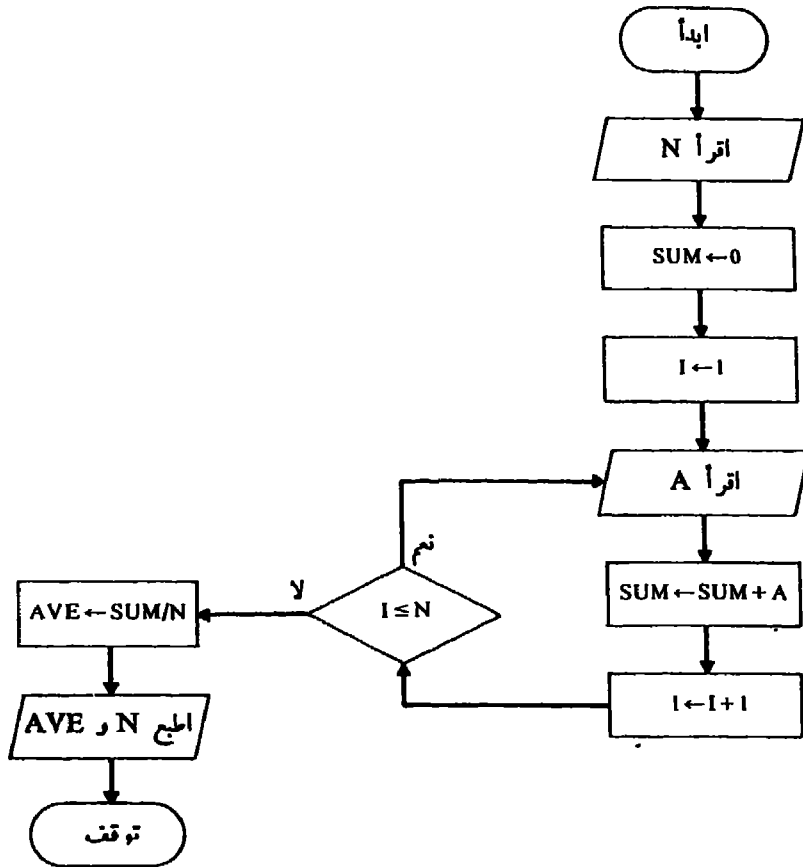


شكل ٤ - ١٩

٤ - ١٠ بطاقة المقدمة والبطاقة الخلفية

البرامج التي تم تطويرها في قسم ٤ - ٩ بها عيب واحد ، فجميعها سوف تعمل كما يجب إذا كان هناك مجموعة من 100 بطاقة فقط . يجب أن يمدل البرنامج إذا كانت المجموعة المعطاة تتكون من 76 بطاقة ، مع أن النظام الحسابي (الخوارزم) نفسه لا يتغير ومن الواضح أن المطلوب هو كتابة برنامج لا يعتمد على عدد البطاقات الموجودة في مجموعة البيانات ، أي أن المطلوب كتابة برنامج يعمل لأي عدد من بطاقات البيانات سواء كانت 4 أو 100 أو 76 . وسوف نناقش طريقتين يمكننا بهما أن نتجز ذلك : (١) استخدام بطاقة المقدمة و (٢) استخدام البطاقة الخلفية .

في خريطة سير العمليات في شكل ٤ - ١٩ (ب) ، نجد أن الحلقة التكرارية تتكرر 100 مرة وذلك لإيجاد مجموع 100 قيمة . افترض ، أننا نريد أن نكرر الحلقة التكرارية لعدد متغير من المرات (تباعاً لمسألة معينة) وعليه يجب أن نسمي هذا العدد باسم متغير وليكن N . لتنفيذ البرنامج ، مع استخدام المتغير N ببدلاً من العدد 100 يجب أن نعرف قيمة N عند وقت التنفيذ . ويمكن أن يتم ذلك بتتقيب عدد بطاقات المجموعة على بطاقة تسمى بطاقة المقدمة وتوضع أعلى مجموعة البيانات . ويقراءة هذا العدد يتم تعريف N وخريطة سير العمليات لهذا البرنامج معطاة في شكل ٤ - ٢٠ .



شكل ٤ - ٢٠

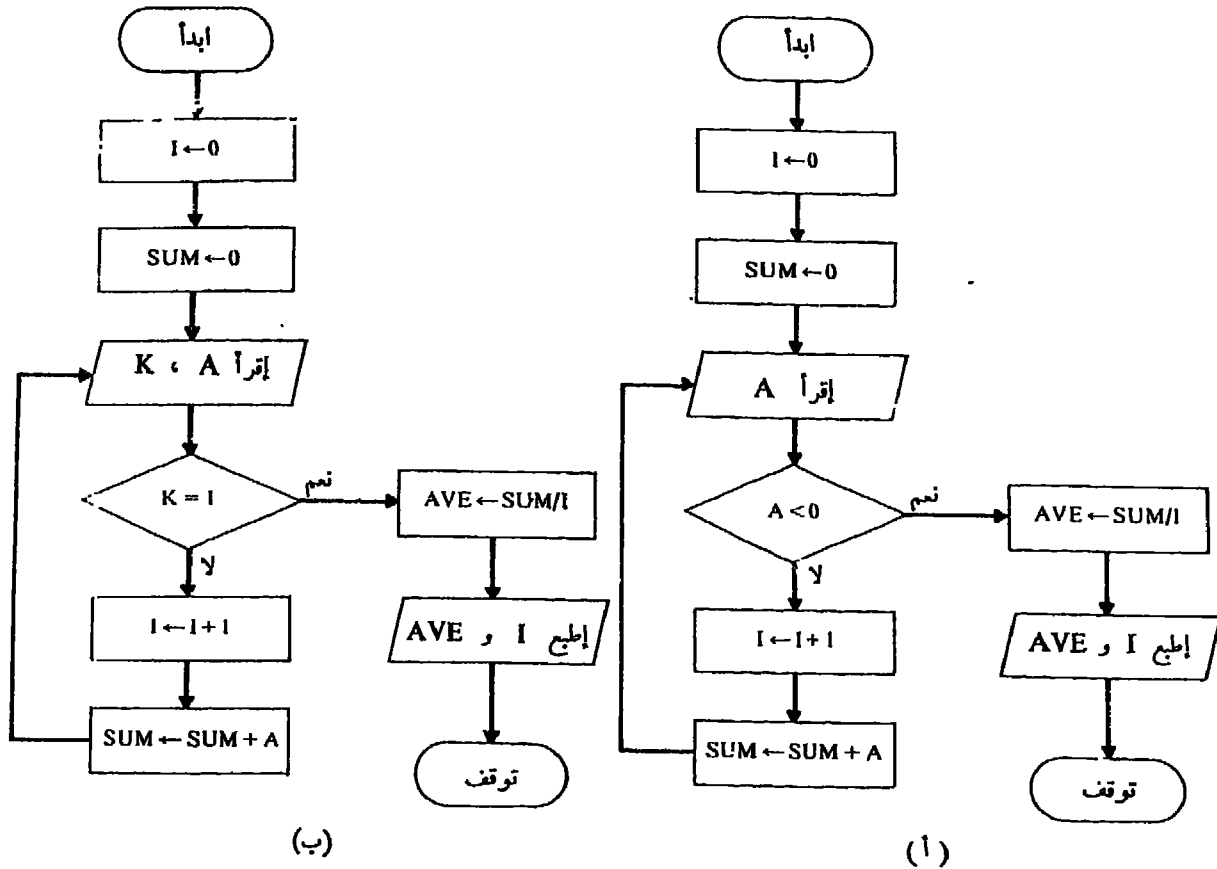
يصعب أحياناً استخدام بطاقة المقدمة فثلاً إذا كان المطلوب حساب متوسط درجات الامتحان لعدد من الطلبة يتراوح ما بين 200 و 300 طالب فلنكي نمد بطاقة المقدمة ، يجب أولاً أن يتم تحديد عدد بطاقات البيانات في المجموعة . وبما أن عدد الطلبة يتراوح ما بين 200 و 300 طالب ، فعدد البطاقات سيتراوح بين 200 و 300 بطاقة . ويستحسن بالطبع أن يقوم الحاسب بعملية المد . ويمكن أن يتم هذا باستخدام البطاقة الخلفية ، التي نناقشها فيما يلي .

افترض أننا سنحسب متوسط درجات امتحان كما سبق . ولما كان العدد الذي يمثل درجة الامتحان يقع ما بين صفر و 100 فقط ، فنستطيع أن نضع في آخر المجموعة بطاقة إضافية تحتوي على رقم خارج هذا النطاق : كاستخدام عدد سالب مثلاً . وتسمى هذه البطاقة البطاقة الخلفية . وحين يقابل الحاسب عدداً سالباً ، يعطى إشارة بأن مجموعة البيانات قد إنتهت وخريطة سير العمليات لمثل هذا البرنامج تظهر في شكل ٤ - ٢١ (أ) .

ملاحظة : لاحظ أنه في هذا البرنامج أعطى العداد قيمة ابتدائية صفرأ . والسبب في ذلك أننا نريد أن نعرف عدد الأعداد الموجبة فقط ، إذ أننا لا نعرف أن العدد موجب إلا بعد أن يتم اختياره . وبذلك ، فقيمة I بعد تنفيذ $I = I + 1$ تحدد آخر بطاقة تمت قراءتها وليس البطاقة الجاري قراءتها .

يعمل بدسلوب السابق فقط عندما نعرف مدى الأعداد ، إلا أنه بعد إجراء تغيير فيه يمكن استخدامه حتى إن لم تكن هناك أي معرفة عن البيانات التي سيتناولها ، يمكن أن تقرأ كأصفار (في الحقول الرقية) فالحاسب يشعر بنهاية مجموعة البيانات حين تأخذ K القيمة I وتظهر خريطة سير العمليات لهذا البرنامج البديل في شكل ٤ - ٢١ (ب) .

أخيراً نشير إلى أنه في بعض الاحيان يطلق على بطاقات المقدمة والبطاقات الخلفية البطاقات الحارسة .



شكل ٤ - ٢١

مسائل محلولة

جبل IF

٤ - ١ اكتب الجمل الآتية بالفورتران

- (أ) If $X > Y$, stop.
 (ب) If $J \neq K$, go to the statement labeled 31.
 (ج) If $A^2 \leq B + C$, go to the statement labeled 41.
 (د) If $A - B \geq X^3$, stop.

IF(A**2.LE.B + C) GO TO 41 (ج)
 IF(A - B.GE.X**3) STOP (د)

IF(X.GT.Y) STOP (أ)
 IF(J.NE.K) GO TO 31 (ب)

٤ - ٢ افترض أنه تم تعريف X و Y . اكتب جملة فورتران أو جزءاً من برنامج فورتران الذي (أ) ينقل التحكم إلى الجملة رقم 41 إذا كانت $X^2 \leq Y$ وينقل التحكم إلى الجملة رقم 42 إذا كانت غير ذلك ، (ب) يضع $K = 0$ إذا كانت $X + Y > 100$ ويضع $K = 1$ إذا كانت غير ذلك .

نقد ماسبق بطريقتين ، مرة بجملة IF الحسابية ، ومرة أخرى بجملة IF المنطقية .

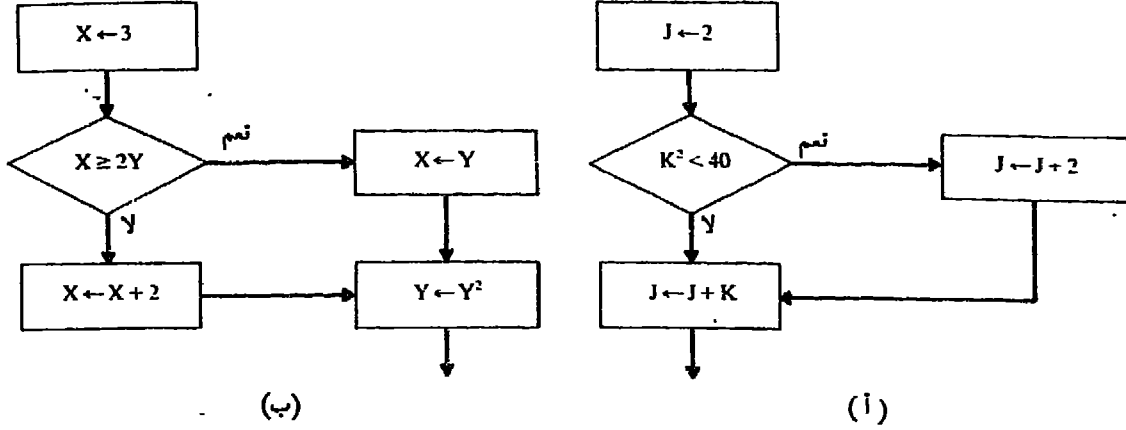
(أ) (١) لاحظ أن الشرط $X^2 \leq Y$ يتحقق إذا كانت قيمة $X^2 - Y$ سالبة أو صفراً فقط .

```

IF(X**2 - Y) 41, 41, 42
IF(X**2.LE.Y) GO TO 41
42 .....
(ب) (١) لاحظ أن الشرط  $X + Y > 100$  يتحقق فقط إذا كانت قيمة  $X + Y - 100$  موجبة .
IF(X + Y - 100.0) 20, 20, 10
10 K = 0
GO TO 30
20 K = 1
30 *****
IF(X + Y.LE.100.0) GO TO 20
K = 0
GO TO 30
20 K = 1
30 *****

```

٤ - ٣ ترجم خريطتي سير العمليات في الشكل ٤ - ٢٢ إلى أجزاء برنامج فورتران .



شكل ٤ - ٢٢

(أ) الخريطة تكافئ: IF... THEN... كما سبق وأن ناقشناها في حالة (١) من قسم ٤ - ٥ .

```

J = 2
IF(K**2.LT.40) J = J + 2
J = J + K
  
```

(ب) الخريطة تكافئ: IF... THEN... ELSE كما سبق وأن ناقشناها في حالة (٢) من قسم ٤ - ٥ .

```

X = 3.0
IF(X.GE.2*Y) GO TO 10
X = X + 2.0
GO TO 20
10 X = Y
20 Y = Y**2
  
```

٤ - ٤ افترض أنه تم تخزين T_1 و T_2 و T_3 في الذاكرة وترمز هذه المتغيرات إلى درجات اختبار . اكتب جزءاً من برنامج الفورتران الذي يحيد ويعطي عدد الدرجات التي تساوى أو تزيد عن 90 (مسألة أكثر شمولاً معطاة في ٤ - ١٤) .

يجعل N ترمز إلى عدد درجات أكبر من أو تساوى 90 (لذا ، N يجب أن تساوى 0 أو 1 أو 2 أو 3)

```

N = 0
IF(T1.GE.90.0) N = N + 1
IF(T2.GE.90.0) N = N + 1
IF(T3.GE.90.0) N = N + 1
WRITE(6, 10) N
10 FORMAT(1X, I5)
  
```

٤ - ٥ اكتب جزءاً من برنامج فورتران باستعمال جملة IF الحاسوبية التي تكافؤ كلا مماز يآ ز جمل S و T هنا قابلة للتنفيذ)

$$\text{IF}(J.NE.K) S \quad \text{(ب)} \quad \text{IF}(A.LT.B) S \quad \text{(أ)}$$

T

T

(أ) لاحظ أن الشرط $(A < B)$ يتحقق فقط إذا كانت $(A - B)$ سالبة

IF(A - B) 10, 20, 20
10 S
20 T

(ب) لاحظ أن الشرط $(J \neq K)$ يتحقق فقط إذا كانت $(J - K)$ سالبة

IF(J - K) 10, 20, 10
10 S
20 T

٤ - ٦ افترض أن J تحتوي على 5 و K تحتوي على 10 . اوجد القيمة النهائية ل J بعد كل جزء من برامج الفورتران التالية :

IF(4*J - 2*K) 10, 20, 20 (د) IF(3*J.LT.K) J = J + 2 (أ)
10 J = K J = J + 3

20 J = J + 1

IF(2*K.LE.3*J) GO TO 50 (هـ) IF(2*J.EQ.K) J = J + 2 (ب)
J = J + 1 J = J + 3
GO TO 60

50 J = K

60 J = J + K

IF(K.GT.J) GO TO 50 (و) IF(K - J) 10, 20, 10 (ز)
J = J + 1 10 J = K

GO TO 60 20 J = J + 1

50 J = K

60 J = J + K

(أ) حيث أن $3J < K$ غير صحيحة فإن $J = J + 3$ هي التي تنفذ ، لذا فإن قيمة J النهائية هي 8 .

(ب) حيث أن $2J = K$ صحيحة فإن $J = J + 2$ تنفذ أولاً لتعطي $J = 7$. ثم تنفذ بعد ذلك $J = J + 3$ ؛ ومن ثم تكون قيمة J النهائية هي 10 .

(ج) حيث أن $K - J$ موجبة فتنفذ $J = K$ أولاً لتعطي $J = 10$. ثم تنفذ بعد ذلك $J = J + 1$ التي تعطي قيمة نهائية $J = 11$.

(د) حيث أن $4J - 2K$ هي صفر فتنفذ فقط $J = J + 1$ ، ومن ثم تكون قيمة J النهائية هي 6 .

(هـ) حيث أن $2K \leq 3J$ غير صحيحة ، فتنفذ أولاً $J = J + 1$ لتعطي $J = 6$. ثم تنفذ بعد ذلك $J = J + K$ ، والتي تعطي قيمة نهائية $J = 16$.

(و) حيث أن $K > J$ صحيحة فتنفذ أولاً $J = K$ لتعطي $J = 10$. ثم تنفذ بعد ذلك $J = J + K$ التي تعطي قيمة نهائية $J = 20$.

جملة GO TO المحسوبة

٤ - ٧ اكتشاف الأخطاء ، إن وجدت ، في كل جملة من جملة GO TO المحسوبة .

(أ) GO TO (5, 8, 4) MARK
(ب) GO TO (5, 22, 22, 57), KKK
(ج) GO TO (2, 84, 578), ERIK
(د) GO TO (5, 76, 0, 24), J

(أ) يجب أن تكون هناك فصلة قبل MARK

(ب) لا توجد أخطاء .

(ج) يجب أن نكون ERIK متغيراً صحيحاً .

(د) لا يمكن أن يكون الصفر رقم جملة .

٤ - ٨ أو - د رقم الجملة التي ينتقل إليها التحكم بعد كل جزء من برنامج الفورتران .

(أ) MARK = 3
GO TO (23, 47, 16, 94), MARK
J = 2
J = J + 2
GO TO (23, 47, 16, 94), J
(ب)
(ج) K = 1
K = K + 3
GO TO (23, 16, 94), K

(أ) حيث أن MARK = 3 فإن التحكم ينتقل إلى الجملة رقم 16 وهو الرقم الثالث في القائمة .

(ب) حيث أن J = 4 بعد تنفيذ جملة IF المحسوبة ، لذا ينتقل التحكم إلى الجملة رقم 94 .

(ج) حيث أن K = 4 بعد تنفيذ جملة GO TO المحسوبة ، وحيث أن القائمة تحتوي على ثلاثة أرقام فقط ، ومن ثم ، ستعطي رسالة خطأ .

البرامج

٤ - ٩ ارسم خريطة سير العمليات وأكتب البرنامج الذي يطبع كل عدد فزدي من رقمين N ومرعبة N² وتكعيه N³ بحيث تظهر قيم N المختلفة على أسطر مختلفة . يجب أن يكون البرنامج مكتوباً بحيث يكون للمود N العنوان NUMBER ، وللمود N² العنوان SQUARE وللمود N³ العنوان CUBE .

نجدل N = 11 و N² = 121 و N³ = 1331 تختبر لرى إذا كانت N < 100 وذلك بعد زيادة N بمقدار 2 . خريطة سير العمليات معطاة في الشكل ٤ - ٢٣ (أ) وما يقابلها بالفورتران في شكل ٤ - ٢٣ (ب) .

لو أضفنا الجملتين التاليتين في بداية البرنامج

WRITE(6, 5)
5 FORMAT(7X, 'NUMBER', 3X, 'SQUARE', 5X, 'CUBE')

حينئذ يكون المخرج موضعاً كما في الشكل ٤ - ٢٣ (ج) .

```

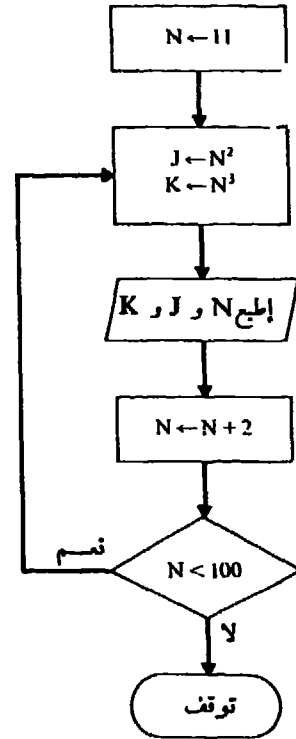
N = 11
10 J = N**2
   K = N**3
   WRITE(6, 20) N, J, K
20 FORMAT(1X, 3I10)
   N = N + 2
   IF(N.LT.100) GO TO 10
   STOP
   END

```

(ب)



(ب)



(أ)

شكل ٤ - ٢٣

٤ - ١٠ ادرم خريطة سير العمليات واكتب برنامج فورتران يحسب التالى (إلى خمسة أرقام عشرية)

$$\frac{2}{1} \cdot \frac{4}{3} \cdot \frac{6}{5} \cdot \dots \cdot \frac{22}{21} \quad (\text{ب})$$

$$\frac{1}{1} + \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{21} \quad (\text{أ})$$

(أ) نضع أولاً $SUM = 0$ ثم نجعل بالتتابع $1/K$ إلى SUM لكل من $K = 1, 3, 5, \dots, 21$ يعطى الشكل ٤ - ٢٤
 (أ) خريطة سير العمليات لاحظ أننا نضع $K = 1$ أولاً ثم نزيد K بمقدار ٢ في كل مرة لأن المجموع يشمل الأعداد الفردية فقط . وعلاوة على ذلك ، نختبر إذا كانت $K \leq 21$ حيث أننا لا نريد أن نجمع أى شيء بعد جمع $1/21$. عند ترجمة خريطة سير العمليات إلى الفورتران ، يجب أن نكتب .

1.0/FLOAT(K)

بدلاً من كتابته $1/K$ حيث أننا لا نريد أن نجري قسمة صحيحة . يظهر البرنامج كما يلي :

```

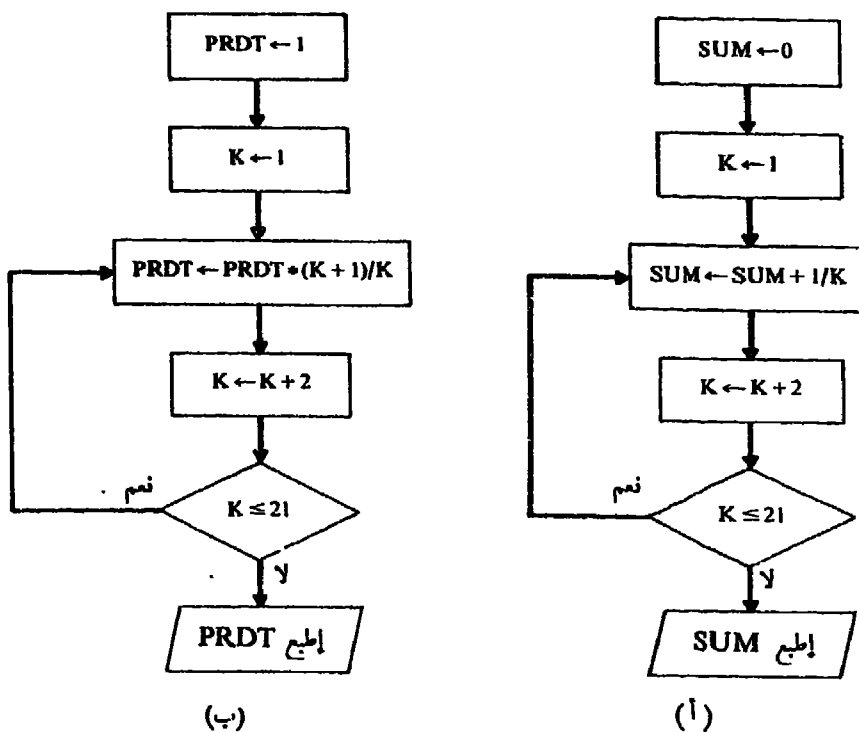
SUM = 0.0
K = 1
11 SUM = SUM + 1.0/FLOAT(K)
   K = K + 2
   IF(K.LE.21) GO TO 11
   WRITE(6, 20) SUM
20 FORMAT(1X, 'THE SUM IS', 2X, F8.5)
   STOP
   END

```

(ب) نجعل $PRDT = 1$ أولاً بعد ذلك نضرب $PRDT$ في $(K + 1)/K$ لكل من $K = 1, 3, \dots, 21$. يبين شكل ٤ - ٢٤ (ب) خريطة سير العمليات المطلوبه لاحظ التشابه مع الجزء (أ) . وفيما يلي البرنامج :

```

PRDT = 1.0
K = 1
13 PRDT = PRDT*FLOAT(K + 1)/FLOAT(K)
K = K + 2
IF(K.LE.21) GO TO 13
WRITE(6, 30) PRDT
30 FORMAT(1X, 'THE PRODUCT IS', 2X, F8.5)
STOP
END
    
```



شكل ٤ - ٢٤

٤ - ١١ إترض أنه تم إيداع مبلغ \$2000.00 في حساب توفير في سنة 1977 وافترض أن البنك يدفع 6 في المائة فائدة مركبة سنوياً على الحساب . اكتب برنامجاً يطبع (السنة) YEAR و (القيمة) AMOUNT لهذا الحساب إلى سنة 1995 .

تزداد AMOUNT كل سنة بمقدار 6 في المائة وبذلك يكون التخصيص

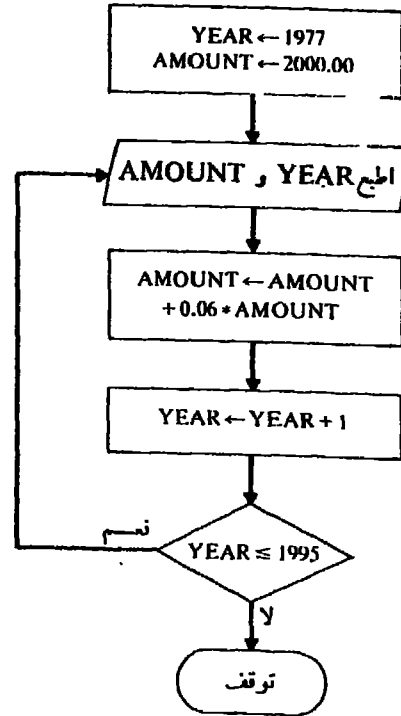
$$AMOUNT \leftarrow AMOUNT + 0.06 * AMOUNT$$

نكرر هذه العملية طالما $YEAR \leq 1995$ يبين شكل ٤ - ٢٥ خريطة سير العمليات للبرنامج ومايقابلها بالفورتران .

```

INTEGER YEAR
WRITE(6, 10)
10 FORMAT('1', 6X, 'YEAR', 6X, 'AMOUNT')
YEAR = 1977
AMOUNT = 2000.00
55 WRITE(6, 20) YEAR, AMOUNT
20 FORMAT(1X, I10, 3X, 'F', F8.2)
AMOUNT = AMOUNT + 0.06*AMOUNT
YEAR = YEAR + 1
IF(YEAR.LE.1995) GO TO 55
STOP
END
    
```

(ب)



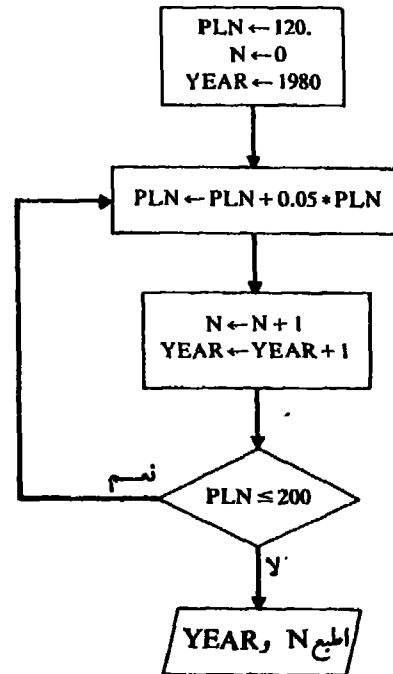
(أ)

شكل ٤ - ٢٥

```

INTEGER YEAR
PLN = 120.0
N = 0
YEAR = 1980
100 PLN = PLN + 0.05*PLN
N = N + 1
YEAR = YEAR + 1
IF(PLN.LE.200) GO TO 100
WRITE(6, 10) N, YEAR
10 FORMAT(1X, I3, 3X, I4)
STOP
END
    
```

(ب)



(أ)

شكل ٤ - ٢٦

٤ - ١٢ افترض أن تعداد بلد PLN في سنة 1980 يبلغ 120 مليون نسمة وافترض أن التعداد يزداد بنسبة 5 في المائة كل سنة . اكتب البرنامج الذي يحدد عدد السنوات N والسنة YEAR عندما يتعدى التعداد 200 مليون نسمة .

يزاد PLN كل سنة بنسبة 5 في المائة . وبذلك فإن التخصيص يتكرر طالما $PLN \leq 200$.

$$PLN \leftarrow PLN + 0.05 * PLN \quad (1)$$

تظهر خريطة سير العمليات للبرنامج في شكل ٤ - ٢٦ (أ) . لاحظ أننا نطبع N و YEAR فقط بعد أن يتعدى PLN الرقم 200 . تظهر ترجمة البرنامج بالفورتران في شكل ٤ - ٢٦ (ب) .

(يمكن أن تكتب $PLN * 1.05$ بدلا من $PLN + 0.05 * PLN$)

٤ - ١٣ افترض أن كل بطاقة من مجموعة البطاقات تحتوي على عدد حقيقي . أضيفت بطاقة مقدمة تحتوي على عدد البطاقات في المجموعة N . -ريد أن نجد أكبر عدد للبطاقات في المجموعة .

(أ) ما هي التغيرات التي يجب عملها في الشكل ٤ - ١٧ لحل المسألة . (يبين شكل ٤ - ١٧ مجموعة بطاقات عددها 100 بطاقة فقط) .

(ب) اكتب البرنامج .

(أ) نحتاج إلى إضافة READ N فقط في بداية خريطة سير العمليات ، وتغيير $99 \leq I$ إلى $I \leq N - 1$ نجا عدا ذلك . ستكون خريطة سير العمليات مطابقة للخريطة السابقة .

(ب) يظهر البرنامج فيما يلي . لاحظ أننا احتجنا لجملة نوع لتعلن أن LAR متغير حقيقي (REAL) .

```

C
C      PROGRAM FINDING LARGEST NUMBER
C
      REAL LAR
      READ(5, 10) N
10     FORMAT(I6)
      READ(5, 20) LAR
      I = 1
100    READ(5, 20) X
      20     FORMAT(F12.2)
      IF(LAR.GE.X) GO TO 200
      LAR = X
200    I = I + 1
      IF(I.LE.N - 1) GO TO 100
      WRITE(6, 30) LAR
30     FORMAT(1X, 'THE LARGEST NUMBER IS', 2X, F12.2)
      STOP
      END

```

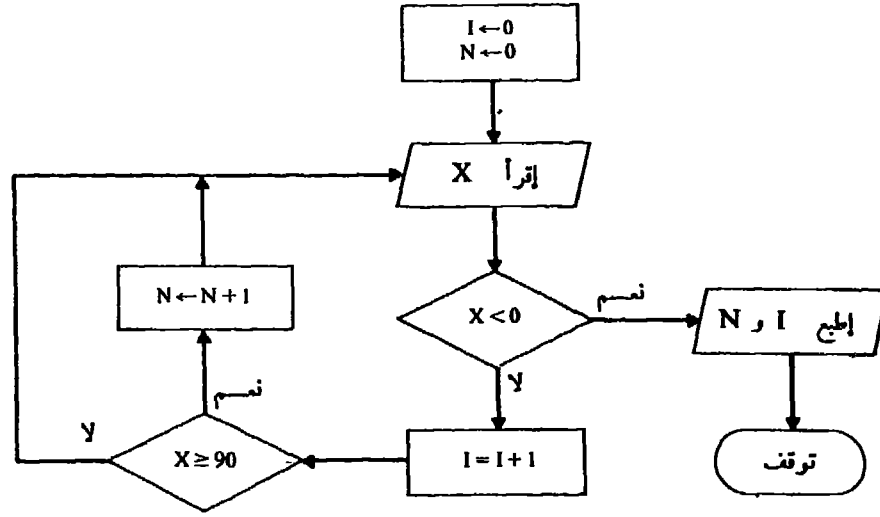
٤ - ١٤ افترض أن كل بطاقة من مجموعة بطاقات تحتوي على درجة اختبار لطالب . وكانت المجموعة لها بطاقة خلفية مثقبة عليها عدد سالب . ارسم خريطة سير العمليات واكتب البرنامج الفورتران الذي يحسب عدد الطلبة I الذين أدوا الاختبار ويحسب أيضاً العدد N وهي عدد درجات الاختبار التي تساوى أو تزيد عن 90 .

شكل ٤ - ٢٧ يبين خريطة سير العمليات . لاحظ أن I أعطيت قيمة ابتدائية صفر . ووجبا إلى مايقابل الخريطة بالصور تران.

```

C
C      PROGRAM SCORES
C
      I = 0
      N = 0
100  READ(5, 10) X
      10  FORMAT(F:0.1)
      IF(X.LT.0.0) GO TO 200
      I = I + 1
      IF(X.GE.90.0) N = N + 1
      GO TO 100
200  WRITE(6, 20) I, N
      20  FORMAT('0', I10, 2X, 'STUDENTS TOOK THE TEST'/
1      '0', I10, 2X 'SCORED ABOVE 90')
      STOP
      END

```



شكل ٤ - ٢٧

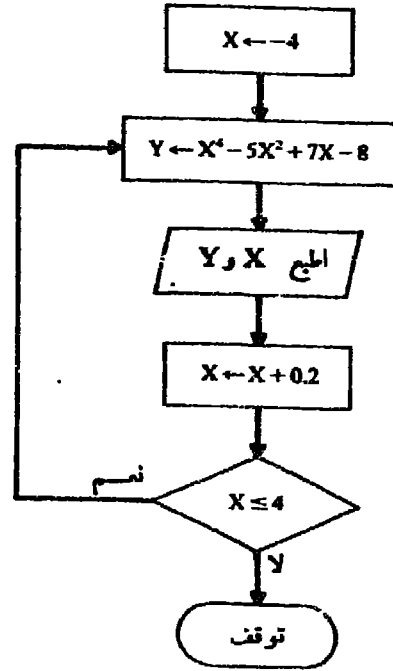
٤ - ١٥ إدرس المعادلة $y = x^4 - 5x^2 + 7x - 8$. ارسم خريطة سير العمليات واكتب برنامج الفورتران الذي يحسب قيمة y لقيم x إبتداء من 4 - إلى 4 بزيادة قدرها 0.2 كل مرة . اطبع قيمة x وقيمة y المناظرة على أسطر مختلفة .

أولا خصص القيمة 4 - إلى X وبمد حساب Y وطباعة Y و X تزيد قيمة X بمقدار 0.2 . وتكرر العملية طالما $X \leq 4$.
تظهر خريطة سير العمليات للنظام الحسابي (الخوارزم) في شكل ٤ - ٢٨ (أ) ويظهر مايقابله بالفورتران في شكل ٤ - ٢٨ (ب) .

```

X = -4.0
100 Y = X**4 - 5.0*X**2 + 7.0*X - 8.0
    WRITE(6, 10) X, Y
10  FORMAT(6X, 'X =', F5.1, 3X, 'Y =', 2X, F6.2)
    X = X + 0.2
    IF(X.LE.4.0) GO TO 100
    STOP
    END
    
```

(ب)



(أ)

شكل ٤ - ٢٨

مسائل تكميلية

جمل IF

٤ - ١٦ اكتب الجمل التالية بالفورتران

- (أ) توقف إذا كانت $A > B$
- (ب) اذهب إلى الجملة رقم 20 إذا كانت $J = K + 3$
- (ج) اذهب إلى الجملة رقم 30 إذا كانت $A + B^2 < 100$
- (د) اذهب إلى الجملة رقم 40 إذا كانت $X - Y \geq 50$
- (هـ) اذهب إلى الجملة رقم 50 إذا كانت I يحوي 4
- (و) توقف إذا كانت $J \leq K$

٤ - ١٧ اكتشف الأخطاء ، إن وجدت في كل من الجمل الآتية :

- | | | | |
|----------------------------|-----|---------------------------|-----|
| IF(X.LE.100) GO TO K | (أ) | IF(A = B) GO TO 50 | (أ) |
| IF(A - 100) 10, 20, 30, | (ب) | IF(X GT Y) STOP | (ب) |
| IF(X.GE.Y) GO TO 55 | (ج) | IF(B**2 - A*C) STOP | (ج) |
| IF(INTEREST.LT.AMOUNT STOP | (د) | IF(X.LT.Y + Z) 10, 15, 20 | (د) |

٤ - ١٨ نفذ كلا ما يأتي بطريقتين ، مرة باستخدام جمل IF المنطقية ومرة أخرى باستخدام جمل IF الحسابية .

(أ) : ينقل التحكم إلى الجملة رقم 100 إذا كانت $A + B^2 > 100$ وانقل التحكم إلى الجملة رقم 200 فيما عدا ذلك .

(ب) : ضع $K = 1$ إذا كانت $A - B \leq 100$ وضع $K = 2$ فيما عدا ذلك .

(ج) : انقل التحكم إلى الجملة 10 أو 20 أو 30 تبعاً للشروط $J = K$ و $J > K$.

٤ - ١٩ أعد كتابة الآتي : باستخدام جمل IF الحسابية . (هنا S و T جمل قابلة للتعميد)

IF(X.LE.Y) S (ج)
T
IF(A.GE.B) S (د)
T

IF(X.GT.Y) S (أ)
T
IF(I.EQ.J) S (ب)
T

٤ - ٢٠ افرض أن J و K تحتويا 3 و 5 على الترتيب . أوجد قيمة J النهائية بعد كل جزء من البرنامج .

IF(J.LT.K - 1) GO TO 10 (د)
J = J + 2
10 J = J + K
IF(J.GE.K + 1) GO TO 10 (هـ)
J = J + 2
10 J = J + K
IF(2*J - K) 10, 10, 20 (و)
10 J = K
20 J = J + 2

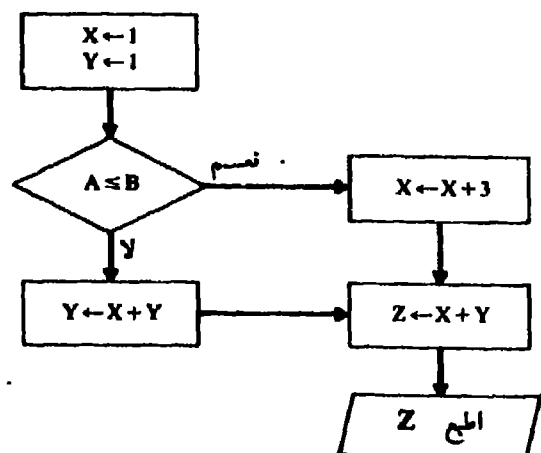
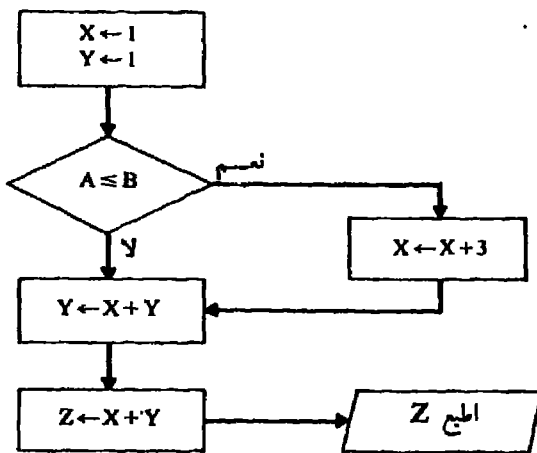
IF(J.GE.K) J = J + 2 (أ)
J = J + 2
IF(5*J.EQ.3*K) J = J + 2 (ب)
J = J + 2

IF(J - K) 10, 10, 20
10 J = K
20 J = J + 2

٤ - ٢١ ترجم خريطتي سير العمليات في شكل ٤ - ٢٩ إلى جزء من برنامج فورتران . وأوجد قيمة Z في كل حالة إذا كانت .

(ب) $A = 3.0, B = 2.0$

(أ) $A = 2.0, B = 3.0$

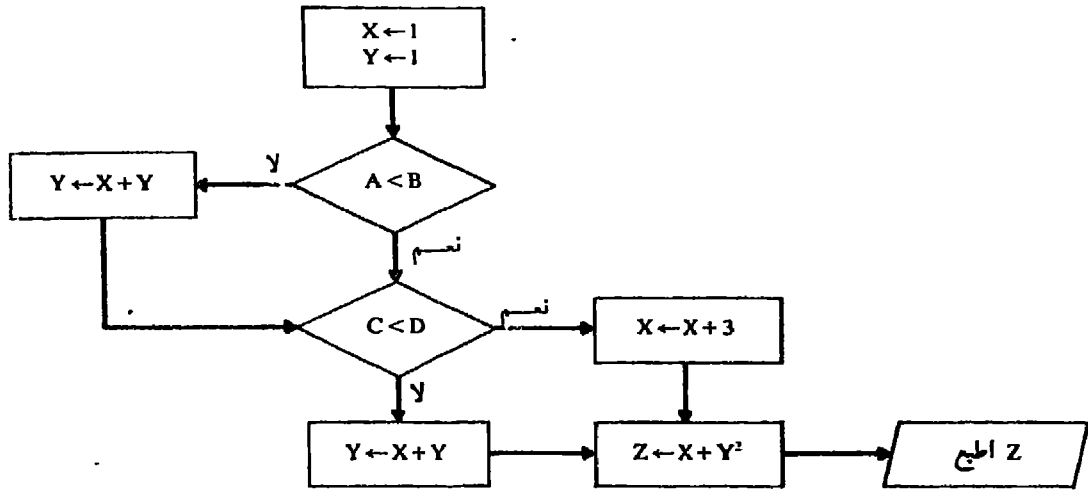


شكل ٤ - ٢٩

٤ - ٢٢ ترجم خريطة سير العمليات في شكل ٤ - ٣٠ إلى جزء من برنامج فورتران ثم أوجد قيمة Z إذا كانت :

(أ) $A = 3.0, B = 2.0, C = 2.0, D = 3.0$
 (ب) $A = 2.0, B = 3.0, C = 3.0, D = 2.0$
 (ج) $A = 3.0, B = 2.0, C = 3.0, D = 2.0$
 (د) $A = 2.0, B = 3.0, C = 2.0, D = 3.0$

(أ) $A = 2.0, B = 3.0, C = 3.0, D = 2.0$
 (ب) $A = 3.0, B = 2.0, C = 3.0, D = 2.0$



شكل ٤ - ٣٠

جمل GO TO المحسوبة

٤ - ٢٣ اكتشف الأخطاء ، إن وجدت في كل جملة من جمل GO TO المحسوبة التالية :

- | | |
|-----------------------------------|------|
| GO TO (35, 17, 17, 46) JIM | (أ) |
| GO TO (42, 0, 88, 1234), LAST | (ب) |
| GO TO (34, 34, 58, 58, 34), N237K | (ج) |
| GO TO (234, 2345678, 7654), J | (د) |
| GO TO (34, 82, 56, 77), TOM | (هـ) |

٤ - ٢٤ اكتب مجموعة من جمل IF المطلقة المكافئة لـ :

GO TO (47, 33, 55, 77), K

٤ - ٢٥ اكتب جملة IF حاسوبية مكافئة لـ :

GO TO (20, 30, 40), JIM

٤ - ٢٦ أوجد رقم الجملة التي ينتقل إليها التحكم بعد كل جزء من برنامج الفورتران التالية :

INTEGER TYPE (أ) $J = 3$ (١)
 $K = 3$
 $TYPE = 2 + K$
 $GO TO (21, 31, 41, 51), J$

INTEGER TYPE (ب) $I = 2$ (ب)
 $K = 2$
 $TYPE = 2 * K$
 $GO TO (21, 31, 41, 51), NEXT$

برامج :

٤ - ٢٧ اكتب البرنامج الذي يطبع الأعداد الصحيحة الموجبة من 1 إلى 300 بحيث تظهر ثلاثة أعداد في كل سطر ، أى ، لكي يبدو المخرج كما يلي :

```

1      2      3
4      5      6
.....
298    299    300

```

٤ - ٢٨ اكتب البرنامج الذي يحسب ، إلى مكانين عشريين ، المجموع التالي

$$1/2 + 2/3 + 3/4 + \dots + 99/100$$

٤ - ٢٩ اكتب برنامجاً الذي يقرأ عدداً صحيحاً موجباً N حيث $N \geq 10$ ثم يحسب المجاميع التالية مقربة إلى مكانين عشريين .

$$1 + 1/2 + 1/3 + \dots + 1/N \quad (أ)$$

$$1 - 1/2 + 1/3 - \dots \pm 1/N \quad (ب)$$

بفرض أن A سبق تعريفها . احسب أيضاً ما يلي

$$\frac{1}{1+A} + \frac{1}{1+2A} + \frac{1}{1+3A} + \dots + \frac{1}{1+N \cdot A} \quad (ج)$$

٤ - ٣٠ اكتب برنامجاً يقرأ عدداً صحيحاً موجباً N حيث $N \geq 10$ ثم يحسب حاصل الضرب التالي مقرباً إلى خمسة أرقام عشرية :

$$\frac{1}{1^2} \cdot \frac{3}{2^2} \cdot \frac{5}{3^2} \cdot \dots \cdot \frac{2N-1}{N^2}$$

٤ - ٣١ اكتب برنامج فورتران لحساب إيجال الدخل GROSS إذا كان كل من ID (رقم الموظف) ، RATE (الأجر في الساعة) و HOUR (عدد الساعات التي عملها الموظف في أسبوع) مخزنة في الذاكرة ، مع ملاحظة أن حيث الأجر الإنساني يدفع على أساس مرة ونصفاً من الأجر .

(الأجر الإنساني هو ما زاد عن ٤٠ ساعة)

٤ - ٣٢ تحسب عمولة المتدوب على المبيعات الإجمالية SALES كالتالي :

(أ) لا تكون هناك عمولة ، إذا كانت $SALES < \$50$

(ب) العمولة = ١٥٪ من المبيعات SALES إذا كانت $\$50 \leq SALES \leq \500

(ج) العمولة = $\$50 + 8\%$ من المبيعات إذا كانت المبيعات تزيد على $\$500$. اكتب البرنامج الذي يقرأ المبيعات الكلية ومنها يحسب العمولة .

٤ - ٣٣ افترض أننا أردنا AMOUNT يساوي $\$1,000.00$ في حساب توفير بنسبة ربح مركبة مقدارها 7 في المائة سنوياً .

(أ) اكتب البرنامج الذي يطبع AMOUNT لحساب كل سنة ولمدة 20 سنة .

(ب) اكتب البرنامج الذي يحدد السنوات التي يستغرقها الحساب حتى تتعدى AMOUNT المقدار $\$5,000.00$.

٤ - ٣٤ افترض أننا قرأنا AMOUNT لوديعة ، بفائدة (مركبة سنوياً) RATE وعدد سنين N . اكتب البرنامج الذي يطبع قيمة الحساب VALUE كل سنة لعدد N من السنوات (إختبر البرنامج مع البيانات في المسألة ٤ - ٣٣) .

٤ - ٣٥ تعرف سلسلة فيبوناتس كما يلي :

$$1, 1, 2, 3, 5, 8, 13, 21, \dots$$

ومنها نجد أن الحدين الأولين لهما القيمة 1 وكل حد بعد ذلك هو مجموع الحدين السابقين :

$$1 + 1 = 2, 1 + 2 = 3, 2 + 3 = 5, 3 + 5 = 8, \dots$$

(أ) اكتب البرنامج الذي يطبع سلسلة فيبوناتس دون أن يتجاوز 10,000

(ب) اكتب البرنامج الذي يطبع أول 50 حد من سلسلة فيبوناتس .

٤ - ٣٦ اجمل J و K أعداداً صحيحة موجبة بحيث تكون $J \leq K$. وتعرف صورة سلسلة فيبوناتس العامة بإعتبار J الحد الأول ، K الحد الثاني ، وكل حد بعد ذلك هو مجموع الحدين السابقين . اكتب البرنامج الذي يقرأ J و K ويطبع أول 50 حد من سلسلة فيبوناتس العامة .

٤ - ٣٧ افترض أن كل بطاقة في المجموعة تحتوي على عدد حقيق موجب (قيمته أقل من 1000) والمجموعة لها بطاقة خلفية .

(أ) اكتب البرنامج الذي يجد العدد الأكبر في المجموعة (قارن مع المسألة ٤ - ٣) ، والتي تشمل بطاقة مقدمة .

(ب) اكتب البرنامج الذي يجد أصغر رقم في المجموعة .

(ج) اكتب البرنامج الذي يجد الرقم الذي يلي أكبر رقم في المجموعة (اعتبر أن المجموعة بها أكثر من رقم واحد) .

٤ - ٣٨ ترجم شكل ٤ - ١٩ (ب) إلى الفورتران . والشكل يحسب المتوسط الحسابي لمائة عدد .

٤ - ٣٩ ترجم شكل ٤ - ٢١ (أ) إلى الفورتران ، والشكل يحسب المتوسط الحسابي لمجموعة من الأعداد غير السالبة . المجموعة لها بطاقة خلفية مشقبة بها عدد سالب .

- ٤ - ٤٠ افرض أن كل بطاقة في مجموعة تحتوي على الدرجة الصحيحة لطالب في اختبار . (أضيفت بطاقة خلفية تحتوي على عدد سالب)
 خصص، لدرجة الاختبار T المتغيرات A أو B أو C أو D و F تبعاً للآتي :
- $T \geq 90$ أو $80 \leq T < 90$ أو $70 \leq T < 80$ أو $60 \leq T < 70$ أو $T < 60$ على الترتيب ، اكتب البرنامج الذي يوجد :
- (أ) عدد التلاميذ الذين أدوا الاختبار (I) .
 (ب) عدد كل من A و B و C و D و F .
 (ج) عدد الأوراق الدرجات النهائية ، أي العدد K من الأوراق الحاصلة على 100 درجة . (المسألة العامة ٤ - ١٤) .

- ٤ - ٤١ افرض ، في الوقت الحالي ، أن تعداد بلدين A و B هو 52 و 85 مليون نسمة على الترتيب ، وأفرض أن معدلات النمو السكاني في A و B هي 6 في المائة و 4 في المائة على الترتيب . اكتب البرنامج الذي يطبع التعداد (إلى أقرب ألف) للبلدين A و B كل سنة إلى أن يتجاوز تعداد A تعداد B . ثم أوجد عدد السنوات N التي استغرقتها تعداد A حتى يتجاوز تعداد B

- ٤ - ٤٢ تماماً مثل مسألة ٤ - ٤١ ، ولكن إذا أعطيت بيانات التعداد ومعدلات النمو للبلدين A و B وليكن POPA و POPB و RATEA و RATEB على الترتيب ، على بطاقات بيانات . (اختبر البرنامج ببيانات مسألة ٤ - ٤١) .

- ٤ - ٤٣ (أ) اكتب البرنامج الذي يطبع كل الأعداد الصحيحة الموجبة أقل من 100 مع حذف تلك الأعداد الصحيحة التي تقبل القسمة على 7 . أي :

1, 3, 5, 9, 11, 15, 17, 19, 23, ..., 97, 99

- (ب) أعد كتابة البرنامج السابق حتى يكون هناك أربعة أرقام على كل سطر ، أي لكي يبدو المخرج كما يلي :

1	3	5	9
11	13	17	19
23	25	27	29

(٢)

- ٤ - ٤٤ اكتب البرنامج الذي يقرأ العدد الصحيح الموجب N حيث $10 \leq N \leq 1000$ ثم يطبع ، على صفحة جديدة ، العدد N و كل الأعداد التي تقبل القسمة عليها حتى يظهر المخرج كما يلي (إذا كانت $N = 12$ مثلاً)

DIVISORS of 12

1
2
3
4
6
12

- ٤ - ٤٥ تجهز بطاقة لكل طالب في فصل الدراسات التكميلية توضح عليها سن الطالب والجنس وموقفه من الفصل والحالة الاجتماعية ، وتبين في أكواد كالتالي :

(١) 1 للأنثى و 2 للذكر

(٢) 1 للطالب المستجد ، 2 بالسنة الثانية ، 3 بالسنة الثالثة ، 4 بالسنة الرابعة ، 5 للتخرج ، 6 لغير المسجل بدراسة نظامية .

(٣) 1 للأعزب ، 2 للمتزوج .

تستعمل بطاقة خلفية لإنهاء المجموعة . أكتب برنامجاً لحساب النسب المئوية لكل مما يأتي : (أ) عدد الذكور (ب) عدد الإناث .
(ج) عدد الطلبة الحريجين (د) عدد الطلبة الذين تتجاوز أعمارهم 30 عاماً .

٤ - ٤٦ اكتب برنامجاً لحساب النسب المئوية مستخدماً البيانات المطبوعة في المسألة ٤٥ - السابقة لما يأتي : (أ) عدد الحريجين المتزوجين
(ب) عدد إناث الفصول العليا (السنوات الثالثة والرابعة) (ج) عدد غير المتخرجين فوق 30 سنة .

٤ - ٤٧ كل بطاقة في مجموعة تحتوي على ثلاثة أعداد موجبة A ، B ، C والمجموعة لها بطاقة خلفية .

(أ) اكتب برنامج لتحديد ما إذا كانت A ، B ، C تشكل أضلاع مثلث . إذا كانت الإجابة نعم ، احسب محيط المثلث ،
أما إذا كانت لا ، فاطبع الرسالة "NOT A TRIANGLE" (تلميح : A و B و C يمكن أن تشكل مثلثاً إذا كان
كل ضلع أقل من مجموع الضلعين الآخرين أي إذا كانت $C < A + B$ ، $B < A + C$ ، $A < B + C$)

(ب) اكتب برنامجاً لتحديد ما إذا كانت A ، B ، C تشكل أضلاعاً : (١) لمثلث متساوي الأضلاع (ثلاثة أضلاع متساوية)
(2) لمثلث متساوي الساقين (ضلعان متساويان) (3) لمثلث قائم الزاوية (الوتر $^2 = \text{ضلع}^2 + \text{ضلع}^2$)

٤ - ٤٨ يعطى مثال ٤ - ه البرنامج الذي يحدد ما إذا كان العدد الصحيح $N > 2$ عدداً أولياً ويتم ذلك باختبار إذا كانت N تقبل
القسمة على أي من الأعداد الصحيحة 2 ، 3 ، ... إلى $N/2$. في الواقع إذا كانت N ليست عدداً أولياً ، فإنها
تقبل القسمة على قاسم \sqrt{N} .

(أ) عدل البرنامج بحيث يختبر ما إذا كانت N تقبل القسمة على أي من الأعداد الصحيحة 2 ، 3 ، ... إلى $\sqrt{2}$ فقط

(ب) عدل البرنامج بحيث يختبر ما إذا كانت N تقبل القسمة على 2 أو على أي عدد صحيح مفرد $\sqrt{N} \geq$ فقط

٤ - ٤٩ ترجم الجملة الآتية إلى الفورتران .

```
IF 1 ≤ X ≤ 2 THEN K = 1
ELSE K = 2
```

(تلميح : الشرط $1 \leq X \leq 2$ يعني $1 \leq X$ ، $X \leq 2$ ويمكن أن تنفذ بواسطة جمل IF المنطقية) .

اجابات للمسائل التكميلية المختارة

٤ - ١٦

IF(X - Y.GE.50.0) GO TO 40	(د)	IF(A.GT.B) STOP	(أ)
IF(L.NE.4) GO TO 50	(هـ)	IF(J.EQ.K + 3) GO TO 20	(ب)
IF(J.LE.K) STOP	(و)	IF((A + B**2).LT.100.0) GO TO 30	(ج)

٤ - ١٧ (أ) .EQ. بدلا من =

(ب) .GT. بدلا من .

(ج) $B^2 - AC$ ليست تعبيراً مترابطاً .(د) $X.LT.Y + Z$ ليست تعبيراً حسابياً .

(هـ) 1000 بدلا من 100 إذا كانت X حقيقية .

(و) لا توجد أخطاء ، رغم أننا نفضل $A - 100.0$.

(ز) لا يوجد أخطاء .

(ح) يجب أن تكون هناك أقواس بين AMOUNT و STOP ،

INTREST لما أكثر من ستة حروف ؛ INTEREST و AMOUNT لما أنواع مختلفة .

٤ - ١٨ (أ) IF(A + B**2.GT.100.0) TO TO 100 or IF(A + B**2 - 100.0) 200, 200, 100
 200
 IF(A - B.LE.100.0) GO TO 10 or IF(A - B - 100.0) 10, 10, 50 (ب)
 K = 2 50 K = 2
 GO TO 20 GO TO 20
 10 K = 1 10 K = 1
 20 20
 IF(J.LT.K) GO TO 10 or IF(J - K) 30, 20, 10 (ج)
 IF(J.EQ.K) GO TO 20
 IF(J.GT.K) GO TO 30

IF(X - Y) 10, 10, 20 (د)
 10 S
 20 T
 IF(A - B) 20, 10, 10 (هـ)
 10 S
 20 T

IF(X - Y) 20, 20, 10 (أ) ٤ - ١٩
 10 S
 20 T
 IF(I - J) 20, 10, 20 (ب)
 10 S
 20 T

٤ - ٢٠

٥ (أ) 5 (ب) 7 (ج) 8 (د) 10 (هـ) 5 (و)

٤ - ٢١

٣.0 و 5.0 (ب) ٩.0 و 3.0 (أ)

٢٢ - ٤

5.0 (أ) 10.0 (ب) 8.0 (ج) 5.0 (د)

٢٣ - ٤ (أ) لا توجد فصلة قبل JIM .

(ب) غير مسوح به 0 .

(ج) لا توجد أخطاء .

(د) 2345678 كبيرة أكثر من اللازم لرقم جملة .

(هـ) TOM يجب أن تكون متغيراً صحيحاً .

IF(K.EQ.1) GO TO 47 ٢٤ - ٤

IF(K.EQ.2) GO TO 33

IF(K.EQ.3) GO TO 55

GO TO 77

IF(JIM - 2) 20, 30, 40 ٢٥ - ٤

٢٦ - ٤

41 (أ)

31 (ب)

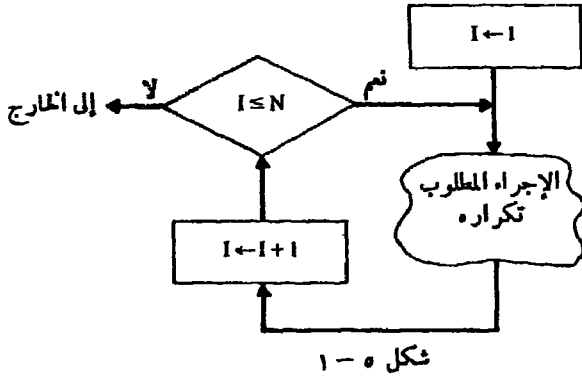
51 (ج)

(د) خطأ حيث أن TYPE تختوى على 5 .

الفصل الخامس

حلقات DO التكرارية

٥ - ١ مقدمة



شكل ٥ - ١

يمكننا نقل التحكم الذي تمت مناقشته في الفصل الرابع من تنفيذ مجموعة من العمليات عدة مرات . نذكر أننا استعملنا عداد I بصيغة جملة IF للخروج من الحلقات التي تمت مناقشتها في القسمين ٤ - ٥ و ٤ - ٧ . توضح خريطة سير العمليات في شكل ٥ - ١ . ميكانيكية التحكم لتكرار إجراء عدد N من المرات . وهي أن العداد I قد أعطى القيمة 1 في البداية . وبعد كل تكرار . تزداد قيمة هذا العداد بمقدار 1 ثم قبل تكرار الحلقة مرة ثانية يسأل هل $I \leq N$? تنهى الدورة بمجرد تجاوز I القيمة N .

وحيث أن تكرار إجراء شيء عملية جوهرية عند كتابة برامج للحاسب ، فن المفيد أن يكون لدينا أمر مطول (macrolike) مثل :

DO...WHILE $1 \leq I \leq N$

DO

$$\left[\begin{array}{c} \vdots \\ \text{الإجراء المطلوب تكراره} \\ \vdots \end{array} \right] \text{ DO } M$$

$$1 \leq I \leq N \text{ ULL}$$

أو أمر مطول مثل DO FOR $1 \leq I \leq N$...REPEAT.

DO FOR $1 \leq I \leq N$

$$\left[\begin{array}{c} \vdots \\ \text{الإجراء المطلوب تكراره} \\ \vdots \end{array} \right] \text{ DO } M$$
 REPEAT

يحتوى الفورتران غير الميكمل على مثل هذا النوع من الأوامر ألا وهي جملة DO . وحيث أن جملة DO هي من أقوى تركيبات الفورتران فقد كرسنا هذا الفصل بالكامل لمناقشة هذا الأمر المطول . (وتسمى بالأمر المطول . أمراً مفرداً يستحضر قائمة من الأوامر التي سبق تحديدها) .

يحتوى الفورتران الميكمل ، الذي يناقش في الفصل الثاني عشر ، على أنواع أخرى من أوامر DO .

٥ - ٢ جملة CONTINUE

نقدم أولاً جملة جديدة قابلة للتنفيذ وهي جملة CONTINUE والتي تتكون ببساطة من الكلمة

CONTINUE

ستتضح أسباب وجود هذه الجملة بعد ذلك في قسم ٥ - ٨ ، ولكن وظيفتها (عملها) تتصف تماماً باسمها - وهي أكل التنفيذ . تسمى أيضاً جملة CONTINUE جملة قابلة للتنفيذ زائفة حيث لا يتولد عنها أمر بلغة الآلة .

تسمح معظم المترجمات باستعمال جملة CONTINUE في أى مكان من البرنامج ، بينما يتطلب البعض أن تحمل جملة CONTINUE رقم .

٥ - ٣ استخدامات بسيطة لجملة DO

تكرار إجراء ما عدد N من المرات ، يمكن أن نستخدم الشكل التالى الشائع الاستخدام لجملة DO .

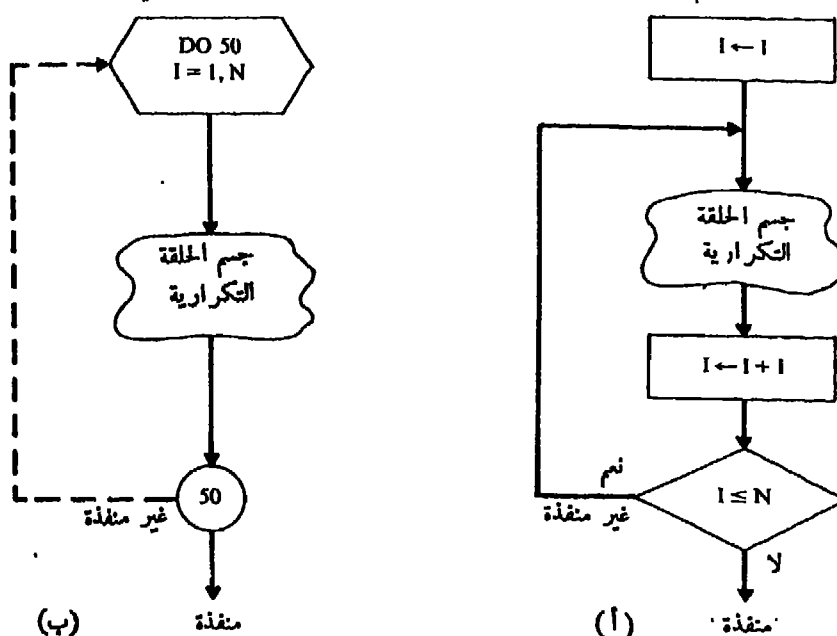
```

DO 50 I = 1, N
    [ BODY OF DO LOOP ]
50 CONTINUE
    
```

يأمر زوج الجمل السابق (DO-CONTINUE) الحاسب أن يكرر تنفيذ الجمل الواقعة بين DO وCONTINUE (وتسمى الحلقة التكرارية) . يخصص للمتغير الصحيح I (ويسمى المتغير الدليل) في البداية القيمة ١ وبعد كل تكرار ، تزداد قيمة I بمقدار ١ . يستمر التكرار طالما $I \leq N$ وعندما تصبح $I > N$ تنفذ الجملة التى تلى جملة CONTINUE بعد ذلك . (نلاحظ أن رقم جملة CONTINUE تم اختياره عشوائياً)

ملاحظة : ولسهولة القراءة ، نحرك عادة جسم الحلقة التكرارية للداخل ولا يؤثر هذا على البرنامج حيث أن المسافات الخالية في الفورتران يتغاضى عنها الحاسب .

توضح خريطة سير العمليات في شكل ٥ - ٢ (أ) المعنى الدقيق لجملة DO السابقة (الغليا) . وحيث أن جملة DO تتضمن عدة صناديق ، فسوف نقدم صندوقاً مطولاً مكانياً ، ذلك بغرض التسهيل كما هو مبين في شكل ٥ - ٢ (ب)

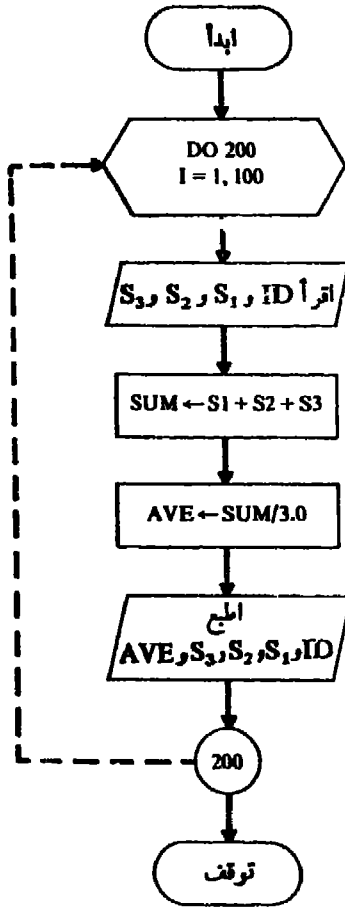


شكل ٥ - ٢

مثال ٥ - ١

تذكر مثال ٤ - ٢ ، الذي يحسب متوسط ثلاث درجات اختبار لكل طالب في فصل به 100 طالب (تتقب درجات اختبار الطالب الثلاث ورقه ID على بطاقة بيانات واحدة) .

وتعيد كتابة البرنامج باستعمال جملة DO ويبين شكل ٥ - ٣ خريطة سير العمليات وما يقابلها في الفورتران . لاحظ كيف يبدو جسم الحلقة التكرارية واضحاً بسبب تحريكه للداخل .



```

WRITE(6, 5)
5  FORMAT('1', 3X, 'ID NO.', 4X, 'TEST 1',
1  4X, 'TEST 2', 4X, 'TEST 3', 3X, 'AVERAGE')
DO 200 I = 1, 100
    READ(5, 10) ID, S1, S2, S3
10  FORMAT(5X, 15, 3(4X, F6.2))
    SUM = S1 + S2 + S3
    AVE = SUM/3.0
    WRITE(6, 20) ID, S1, S2, S3, AVE
20  FORMAT(5X, 15, 4(4X, F6.2))
200 CONTINUE
STOP
END

```

شكل ٥ - ٣

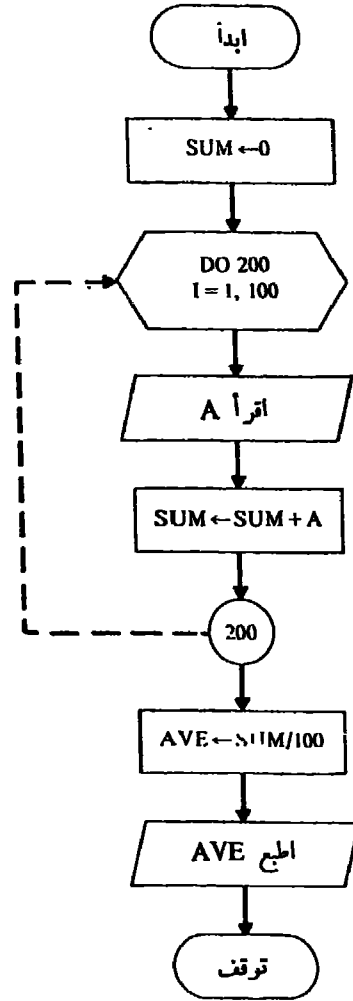
مثال ٥ - ٢

في قسم ٤ - ١٠ من الفصل الرابع ، كتبنا برنامجاً لإيجاد متوسط 100 رقم . وسنعيد كتابة هذا البرنامج باستعمال الصورة البسيطة لحلقة DO .

لاحظ أن الدليل I للحلقة DO في المثال السابق (الأعلى) يمثل عدداً يعد عدد المرات التي مررنا فيها خلال الحلقة . في الواقع ، تكون الحلقة DO أكثر قوة إذا كانت قيمة الدليل I تستخدم أيضاً للحساب بداخل حلقة DO ونوضح هذا في المثال التالي :

```

C
C
C      PROGRAM TO FIND THE AVERAGE OF 100 NUMBERS
      SUM = 0.0
      DO 200 I = 1, 100
        READ(5, 20) A
        FORMAT(F10.2)
20      SUM = SUM + A
200    CONTINUE
      AVE = SUM/100.0
      WRITE(6, 10) AVE
10     FORMAT(1X, F10.2)
      STOP
      END
    
```



شكل ٤ - ٥

مثال ٥ - ٣

(أ) اكتب البرنامج الذي يطبع أول 100 رقم صحيح موجب . (قارن مع قسم ٤ - ٤) تظهر خريطة سير العمليات في شكل ٥ - ٣ (أ) ، وفيما يلي مقابلها بالفورتران :

```

      DO 500 I = 1, 100
        WRITE(6, 10) I
        FORMAT(10X, I3)
10     CONTINUE
500    STOP
      END
    
```

(ب) اكتب البرنامج الذي يحسب مجموع أول 30 رقم صحيح موجب . نريد أن نحسب 30 + + 3 + 2 + 1 أولاً نجعل قيمة ISUM = 0 ثم ننفذ

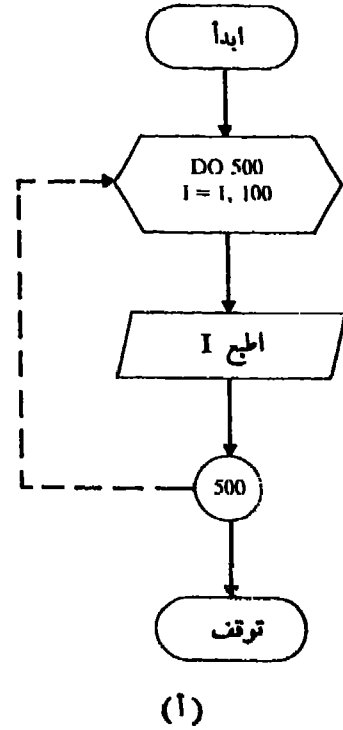
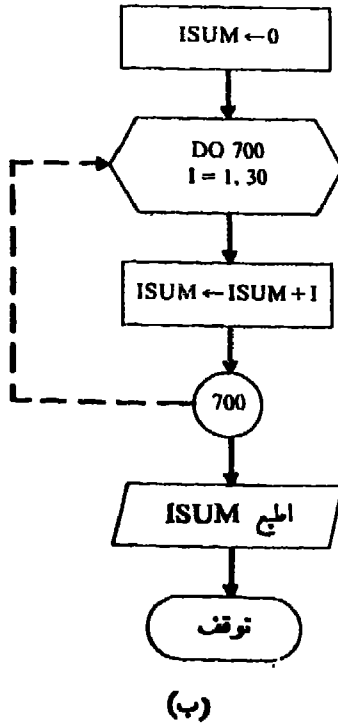
$$ISUM = ISUM + I$$

لكل من 30 - 1, 2, 3, , I باستعمال حلقة DO التكرارية ، تظهر خريطة سير العمليات في شكل ٥ - ٣ (ب) ، وفيما يلي ما يقابلها بالفورتران :

```

ISUM = 0
DO 700 I = 1, 30
    ISUM = ISUM + I
700 CONTINUE
WRITE(6, 20) ISUM
20 FORMAT(10X, I10)
STOP
END

```



شكل ٥ - ٥

(ج) اكتب جزءاً من برنامج يحسب حاصل ضرب أول 10 أرقام صحيحة موجبة . يشبه البرنامج إلى حد كبير ذلك البرنامج في (ب) إلا أننا نريد أن نحسب

1·2·3·...·10

يتم هذا أولاً بحمل قيمة $IPROD = 1$ وننفذ

$IPROD = IPROD * I$

باستعمال حلقة DO التكرارية بدليل I وفيما يلي جزء البرنامج المطلوب .

```

IPROD = 1
DO 100 I = 1, 10
    IPROD = IPROD*I
100 CONTINUE

```

٥ - { جملة DO

حلقة DO التكرارية التي تمت مناقشتها في القسم السابق هي حالة خاصة فقط . وتأخذ حلقة DO التكرارية (غير الميكيلية) العامة الشكل التالي :

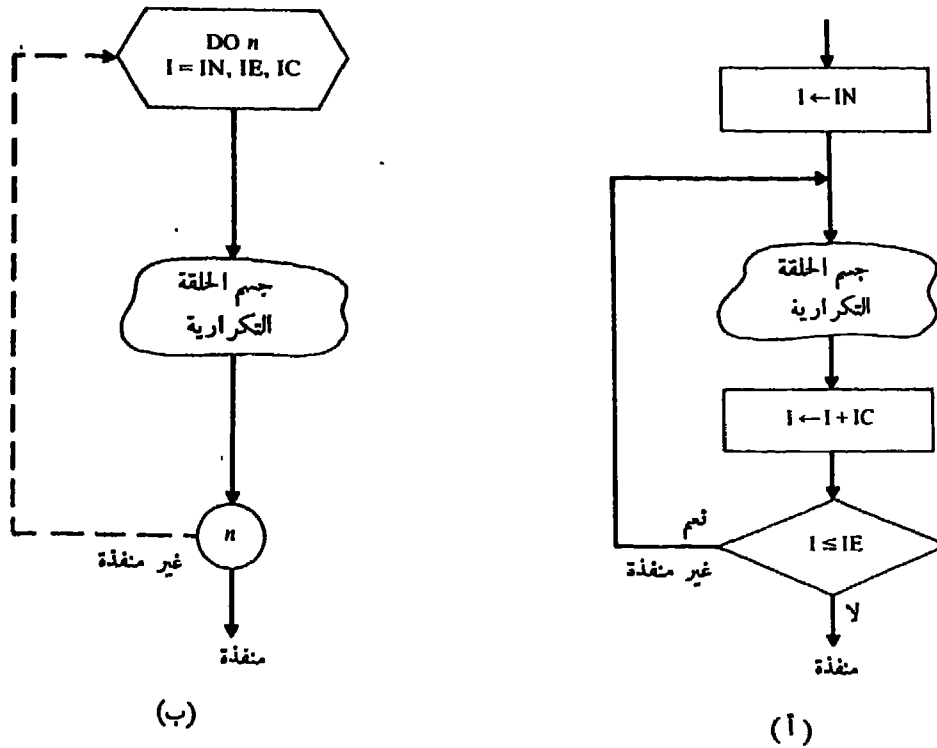
```

DO n I = IN, IE, IC
    [ جسم الحلقة التكرارية ]
n CONTINUE
    
```

ومعاني الرموز n و I و IN و IE و IC هي كالآتي :

- ١- n هي رقم آخر جملة قابلة للتنفيذ في حلقة DO التكرارية . وهي جملة CONTINUE في هذه الحالة ، (انظر قسم ٥-٨) .
- ٢- تشير I إلى اسم متغير صحيح ، وتسمى دليل الحلقة التكرارية .
- ٣- يمكن لكل من IN ، IE ، IC إما أن تكون ثوابت صحيحة موجبة فقط أو أسماء متغيرات صحيحة . ونطلق على هؤلاء معاملات الدليل أو معاملات الحلقة التكرارية . تشير IN إلى القيمة الابتدائية للدليل ، IE هي لقيمة الاعتبار أو قيمة النهاية ، أو القيمة الحدية للدليل ، IC مقدار الزيادة .

ملاحظة : يضم الفورتران الهيكل ، الذي يناقش ، في الفصل الثاني عشر ، أنواع أخرى من حلقات DO التكرارية . أحد هذه الأنواع يشبه الحلقة التكرارية السابقة إلا أنها تسمح بمقدار زيادة سالب وتسمح بقيم ابتدائية وقيم نهائية غير موجبة . تبين خريطة سير العمليات في شكل ٥ - ٦ (أ) حلقة DO التكرارية السابقة (انظر أيضاً الشكل ٥ - ٢) . عندما نقابل جملة DO تحمل قيمة الدليل I مساو للقيمة الابتدائية IN ثم بعد ذلك .



شكل ٥ - ٦

يمر التحكم خلال حلقة DO التكرارية ، أى ، من أول جملة تلى جملة DO حتى جملة CONTINUE . يزداد الآن قيمة الدليل I بمقدار معامل الزيادة IC وتختبر قيمة I الجديدة . إذا تجاوزت I قيمة النهاية IE ، حينئذ تكون حلقة DO التكرارية قد تم تنفيذها : ويمر التحكم إلى أول جملة تلى حلقة DO التكرارية (أى إلى أول جملة بعد جملة CONTINUE) . وإلا فسوف يتنفذ جسم حلقة DO التكرارية مرة ثانية ، وهكذا يأخذ الدليل I قيمة جديدة في كل مرة يمر الحاسب خلال حلقة DO التكرارية . يبين الشكل ٥ - ٦ (ب) خريطة سير العمليات لهذه الحلقة التكرارية ولكن باستعمال الصندوق المطول لحلقة DO التكرارية .

مثال ٥ - ٤

(أ) افرض أن الحاسب قد صادف جملة DO التالية :

```
DO 200 K = 2, 10, 3
```

فيمكن للحاسب أن يدور خلال حلقة DO التكرارية ثلاث مرات ، أولاً عندما $K = 2$ ثم عندما $K = 2 + 3 = 5$ ، وثانياً عندما $K = 5 + 3 = 8$. بعد الانتهاء من الدورات الثلاث تزداد قيمة K إلى $8 + 3 = 11$ ، وفي هذه الحالة تتجاوز قيمة K القيمة النهائية 10 ومن ثم فلا يدور الحاسب خلال حلقة DO التكرارية مرة رابعة . وهذا يوضح أنه ليس من الضروري أن يساوى الدليل قيمة الاختبار (القيمة الحدية) .

(ب) افرض أننا نريد أن نطبع الأعداد الصحيحة الزوجية بين 2 و 100 أى 2, 4, 6, , 100 . نجعل في البداية $IEVEN = 2$ ونفذ $IEVEN = IEVEN + 2$ عدة مرات طالما $IEVEN \leq 100$. وتظهر خريطة سير العمليات في شكل ٥ - ٦ (أ) . وباستعمال حلقة DO التكرارية تنتج خريطة سير العمليات في شكل ٥ - ٦ (ب) ، والمبينة فيما يلي :

```
DO 200 IEVEN = 2, 100, 2
    WRITE(6, 10) IEVEN
10    FORMAT(6X, I3)
200 CONTINUE
```

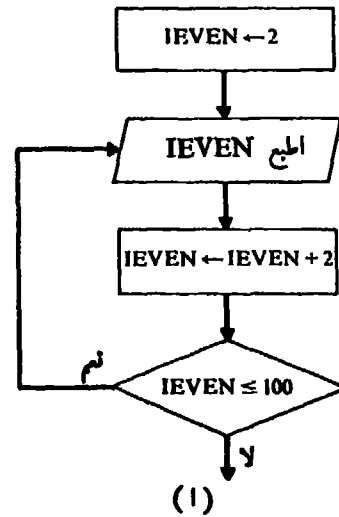
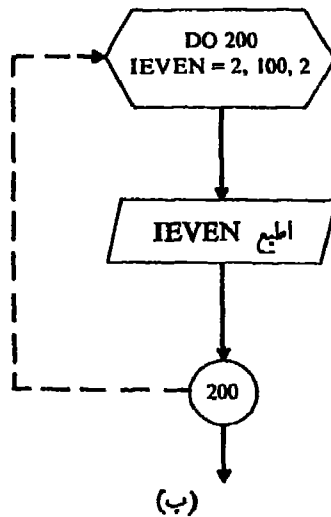
(-) افرض أن A تم تخزينها في الذاكرة . احسب حواصل الضرب .

$$(1 + A)(1 + 2A)(1 + 3A) \dots (1 + 10A)$$

$$(1 + A)(1 + 4A)(1 + 7A) \dots (1 + 19A)$$

وكما في مثال ٥ - ٣ (ج) فيمكن أن يحسب حاصل الضرب (i) باستعمال حلقة DO التكرارية كما يلي :

```
PROD = 1.0
DO 20 I = 1, 10
    PROD = PROD*(1.0 + FLOAT(I)*A)
20 CONTINUE
```



شكل ٥ - ٧

وبتعديل لطيف يمكن أن يحسب حاصل الضرب (**) بالمثل كما يلي :

```
PROD = 1.0
DO 50 I = 1, 19, 3
    PROD = PROD*(1.0 + FLOAT(I)*A)
50 CONTINUE
```

• - • قوانين على استخدام حلقة DO التكرارية

فيما يلي عدد من القوانين تتعلق بحلقة DO التكرارية غير الميكالية السابقة .

١ - إذا كان معامل الزيادة 1 يمكن أن يحذف كلية وعلى سبيل المثال

```
DO 100 I = 1, N, 1
```

يمكن أن تكتب في الشكل المختصر

```
DO 100 I = 1, N
```

٢ - يجب أن تكون كل من معاملات الفهرسة IN ، IE ، IC في أى جملة DO ثابت صحيح موجب ، أو متغير صحيح بدون دليل لقيمة موجبة ، وبذلك فإن

```
DO 200 I = 20, 1, -2
```

غير صحيحة (انظر ملاحظة على صفحة ١٤٠)

تستحق النقطتان التاليتان اهتماماً خاصاً حيث أنهما يقودان إلى أخطاء دائمة .

٣ - لا يمكن القيام بحسابات رياضية في جملة DO نفسها . وبذلك فإن

```
DO 300 I = M, K + 3, 2
```

غير صحيحة ، بينما ، يمكن للفرد أن ينجز نفس الحسابات الرياضية بإدخال متغير جديد ، وليكن KK وكتابة .

```
KK = K + 3
DO 300 I = M, KK, 2
```

٤ - رغم أن الدليل I متاح للحسابات داخل حلقة DO التكرارية لا يجب تغييره بداخل جسم حلقة DO التكرارية . وهذا ينطبق أيضاً بالنسبة إلى معاملات الفهرسة IN ، IE ، IC وبمعنى آخر ، لا يمكن تغيير أى من القيم I ، IN ، IE ، IC بداخل حلقة DO التكرارية (إلا عن طريق ميكانيكية التحكم الملازمة)

مثال • - •

(أ) ادرس جملة DO التالية

```
DO 200 I = 4, 2, 3
```

لاحظ أن القيمة الابتدائية IN تتجاوز القيمة النهائية IE ومع ذلك . ستنفذ حلقة DO التكرارية مرة واحدة ، تبعاً لخريطة سير العمليات (ولكن في بعض المترجمات ، قد يسبب رسالة خطأ)

(ب) افرض أننا نريد أن نطبع الأعداد الصحيحة 100 و 99 و 98 و 1 بهذا الترتيب . فجزء البرنامج التالى الذى يستعمل حلقة DO التكرارية غير الميكلية السابق استخدامها ليس صحيحاً حيث لا يمكن أن يكون معامل الزيادة سالباً .

```
DO 500 I = 100, 1, -1
    WRITE(6, 20) I
20    FORMAT(1X, I3)
500 CONTINUE
```

ومع ذلك ، فإذالك أماننا فرصة أن نكتب مثل هذا البرنامج باستعمال حلقة DO التكرارية غير الميكلية . ونلاحظ أولاً أننا يمكن أن نستخدم الجملة .

```
DO 500 I = 1, 100
```

بينما نريد أن نطبع 100 عندما تكون I هي 1 ونريد أن نطبع 99 عندما $I = 2$ وهكذا . أى

قيمة I : 1 ، 2 ، 3 ، ، 99 ، 100

القيمة التى يجب طباعتها : 100 ، 99 ، 98 ، ، 2 ، 1

ويجمل J تشير إلى القيمة التى نريد طباعتها ، لاحظ أن $I + J$ تساوى دائماً 101 ومن ثم $J = 101 - I$ وبذلك فجزء البرنامج التالى ، الذى يستخدم حلقة DO التكرارية غير الميكلية ، سوف ينجز مهمتنا .

```
DO 500 I = 1, 100
    J = 101 - I
    WRITE(6, 20) J
20    FORMAT(1X, I3)
500 CONTINUE
```

٥ - ٦ الخروج من حلقة DO التكرارية

هناك طريقتان للخروج من أى حلقة تكرارية DO :

(أ) مخرج طبيعى ، (ب) مخرج غير طبيعى . وسوف نناقش ذلك فيما يلى :

(أ) مخرج طبيعى

يحدث المخرج الطبيعى عندما تتجاوز قيمة الدليل I قيمة الاختبار IE (كما هو موضح فى خريطة سير العمليات فى شكل ٥ - ٦ (أ) . فى هذه الحالة ينتقل التحكم إلى أول جملة قابلة للتنفيذ تلى حلقة DO التكرارية صفة واحدة خفية للمخرج الطبيعى من حلقة DO التكرارية هو أن قيمة الدليل I عند وقت الخروج تكون غير معرفة ، ومن ثم لا يجب أن تستعمل فى أى حسابات أخرى . كل الأمثلة التى تمت مناقشتها حتى الآن لما مخرج طبيعى .

(ب) مخرج غير طبيعى

من الممكن الانتقال من داخل إلى خارج حلقة DO التكرارية ، ولكن بواسطة جملة IF فى حلقة DO التكرارية ، حتى إذا لم تتجاوز قيمة الدليل I الحالية قيمة الاختبار IE . سيطلق على مثل هذا المخرج مخرج غير طبيعى . صفة واحدة جوهرية للمخرج غير الطبيعى من حلقة DO التكرارية هو أننا نحفظ بقيمة الدليل I الحالية عند وقت الخروج (من ثم يمكن أن تستخدم هذه القيمة فى أى حسابات أخرى أو فى عمليات I/O)

يوضح هيكل البرنامج التالى ، الذى يستخدم K كدليل لحلقة DO التكرارية متى تكون قيمة الدليل K ، معرفة أو غير معرفة .

```

DO 100 K = 1, 100, 2
  ***
  ***
  IF(X.LT.Y) GO TO 200
  ***
  ***
100 CONTINUE
  *** } قيمة الدليل K غير معرفة بالنسبة لهذه الجمل
  *** }
  *** }
  STOP
100 CONTINUE
  *** } قيمة الدليل K معرفة بالنسبة لهذه الجمل
  *** }
  *** }
  STOP

```

مثال ٥ - ٦

اكتب البرنامج الذي يقرأ عدداً صحيحاً $K > 2$ ويحدد ما إذا كانت K عدداً أولياً أم لا . إن لم تكن ، أعرض المقسوم عليه غير العادي ، ل K (قارن مع مثال ٤ - ٥) لقد برهنا إن لم تكن K عدداً أولياً فإن K لما مقسوم عليه غير عادي $K/2 \geq$. تظهر خريطة سير العمليات والبرنامج في شكل ٥ - ٨ . ونلاحظ أن دليل الحلقة التكرارية يتغير من 2 حتى $K/2$ وحيث لا يمكن إجراء أى حسابات في جملة DO فستستخدم الجملة

$$KK = K/2 \quad \text{أولاً}$$

لاحظ أن حلقة DO التكرارية تختبر كلا من الأعداد الصحيحة 2 ، 3 ، ... ، KK معرفة إذا كان العدد يقبل القسمة على K إن لم يكن ، تستكمل الحلقة التكرارية وبذلك يحدث الفرغ الطبيعي عندما لا يكون أى من الأعداد 2 ، 3 ، ... ، KK قاسماً ل K وبذلك يكون K عدداً أولياً . بينما أى وقت تقبل I القسمة على K يكون لدينا مخرج غير طبيعي . في هذه اللحظة ، لا تزال قيمة الدليل I معرفة ومن ثم يمكن تنفيذ

٥ - ٧ الانتقال بداخل وإلى حلقة DO تكرارية

يمكن دائماً أن تنتقل من أى نقطة في حلقة DO التكرارية إلى نقطة أخرى في نفس الحلقة التكرارية ، ويمكن أن نقفز بحرية إلى خارج أى حلقة DO التكرارية ، إلا أنه ليس ممكناً أن نقفز إلى منتصف حلقة تكرارية ، أى أن الطريقة الوحيدة للانتقال إلى جمل بداخل حلقة DO التكرارية هي عن طريق جملة DO الأصلية . من المهم أن نتذكر أن في كل مرة ينتقل التحكم إلى جملة DO يعاد قيمة دليل الحلقة التكرارية إلى قيمتها الابتدائية .

افرض أننا نريد جزء برنامج فودوران يستخدم حلقة (n) تكرارية التي تحسب المجموع

$$1 + 2 + 3 + 5 + 6 + 7 + 8 + 9 + 10$$

أى مجموع أول 10 أرقام صحيحة موجبة باستثناء العدد الصحيح 4 ادرس جزئى البرنامجين التاليين :

```

ISUM = 0
DO 100 I = 1, 10
    IF(I.EQ.4) I = I + 1
    ISUM = ISUM + I
100 CONTINUE
ISUM = 0
50 DO 200 I = 1, 10
    IF(I.EQ.4) GO TO 50
    ISUM = ISUM + I
200 CONTINUE

```

(أ)

```

ISUM = 0
DO 100 I = 1, 10
    IF(I.EQ.4) I = I + 1
    ISUM = ISUM + I
100 CONTINUE
ISUM = 0
50 DO 200 I = 1, 10
    IF(I.EQ.4) GO TO 50
    ISUM = ISUM + I
200 CONTINUE

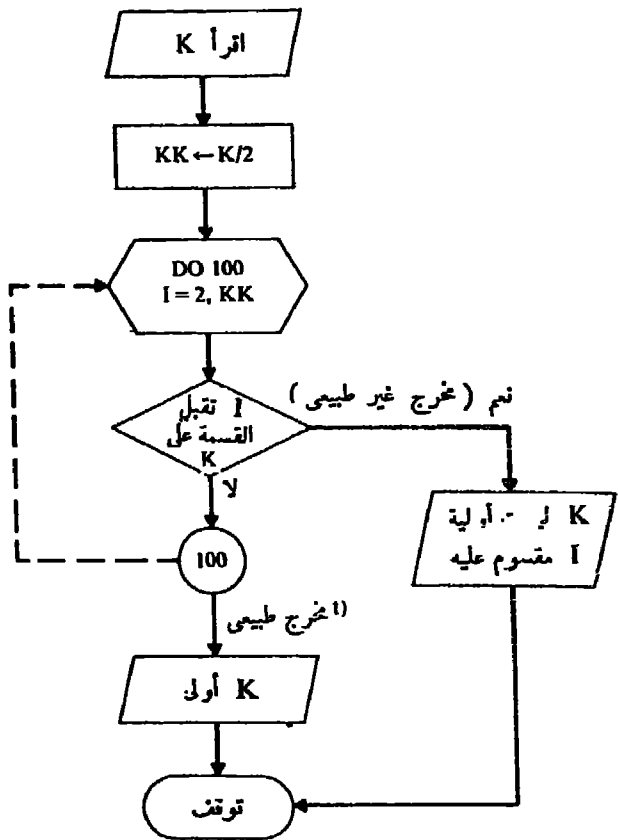
```

(ب)

```

READ(5, 8) K
8  FORMAT(I10)
   KK = K/2
   DO 100 I = 2, KK
       IF(K.EQ.(K/I)*I) GO TO 50
100 CONTINUE
   WRITE(6, 10) K
10  FORMAT(10X, I5, 1X, 'IS A PRIME')
   STOP
50  WRITE(6, 20) K, I
20  FORMAT(10X, I5, 1X, 'IS NOT A PRIME' /
1    10X, I5, 1X, 'IS A DIVISOR')
   STOP
END

```



شكل ٥ - ٨

البرنامج (أ) غير صحيح حيث تم تغيير قيمة الدليل I في حلقة DO التكرارية أي في الجمل

```

IF(I.EQ.4) I = I + 1
ISUM = ISUM + I

```

إن لم تكن موجودة في حلقة DO التكرارية ، إذن فستضاف 5 إلى ISUM إذا كانت I تحتوي أصلاً على 4 . ومع ذلك حيث أننا هنا في منتصف حلقة DO التكرارية فغير مسموح بهذه الجمل .

وعلى النقيض فالبرنامج (ب) يستخدم حلقة DO التكرارية بطريقة صحيحة . ومع ذلك فنطق البرنامج ليس صحيحاً . أي

١٠ - البرمجة بلغة الفورتران

يوجد مخرج غير طبيعي حين تأخذ I القيمة 4 لكن ينتقل التحكم إلى الخلف أي إلى جملة DO وكما ذكرنا من قبل ، كلما أعدنا تنفيذ جملة DO فتعاد قيمة الدليل إلى قيمته الابتدائية التي هي في هذه الحالة 1 وبالتالي سيمطى هذا البرنامج حسابات تكرارية لا نهائية .

$$1 + 2 + 3 + 1 + 2 + 3 + 1 + 2 + 3 + \dots$$

وهكذا

الفرض من البرنامج (ب) هو تحطى $ISUM = ISUM + 1$ عندما تكون I تساوى 4 إلا أننا نريد أيضاً أن نكرر حلقة DO التكرارية في مسارها الطبيعي . ويمكن أن ننجز ذلك بنقل التحكم إلى جملة CONTINUE بدلا من جملة DO أي أن برنامج (ب) سيحسب المجموع المطلوب إذا تغيرت الجملة

```
IF(I.EQ.4) GO TO 50
```

إلى

```
IF(I.EQ.4) GO TO 200
```

يوضح هذا المثال حقيقة النقطة الأساسية في القسم التالى - وهي ضرورة جملة CONTINUE

٥ - ٨ ضرورة جملة CONTINUE

في الحقيقة ليس من الضروري أن تكون جملة CONTINUE هي الجملة الأخيرة في أى حلقة تكرارية DO ، ولكن يجب أن تكون أى جملة قابلة للتنفيذ غير جملة GO TO أو جملة IF الحسابية ، أو جملة DO أخرى . على سبيل المثال فبدلا من

```
ISUM = 0
DO 200 I = 1, 100
    ISUM = ISUM + I
200 CONTINUE
```

يمكن أن نكتب

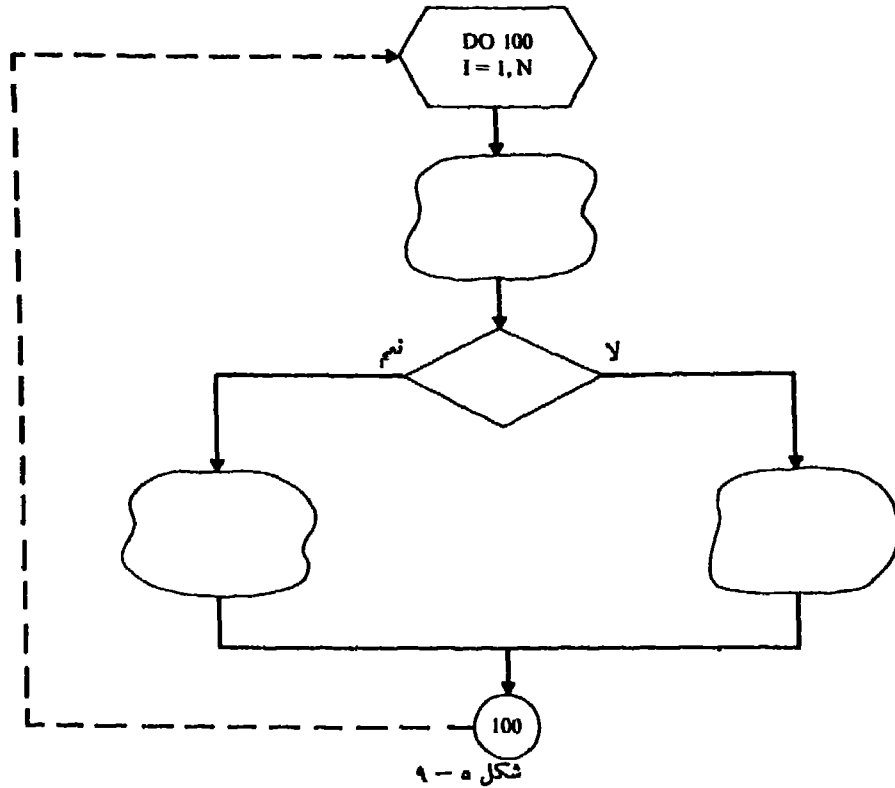
```
ISUM = 0
DO 200 I = 1, 100
200    ISUM = ISUM + I
```

ومع ذلك فما زلنا نحذف استخدام الزوج DO-CONTINUE إذ أنها ستمنع نوع الخطأ الشائع التالى :

```
DO 300 I = 1, 100
    WRITE(6, 300) I
300    FORMAT(10X, I3)
```

(الخطأ هو استعمال جملة FORMAT غير المنفصلة كآخر جملة في الحلقة التكرارية DO فجملة WRITE بدلا من جملة FORMAT هي التي يجب أن تحمل الرقم 300. وأكثر من ذلك أهمية ، يمرض الزوج DO-CONTINUE الميزة التعليمية حيث أنه يقنن شكل أى حلقة تكرارية DO ، وجملة CONTINUE تتحكم كحدود للحلقة التكرارية DO .

ومع ذلك ، هناك موقف يجب أن نستخدم فيه جملة CONTINUE افترض أن هناك جملة شرطية بداخل الحلقة التكرارية DO حيث البديلان ليس بينهما شيء عام ، كما هو موضح في الشكل ٥ - ٩ .

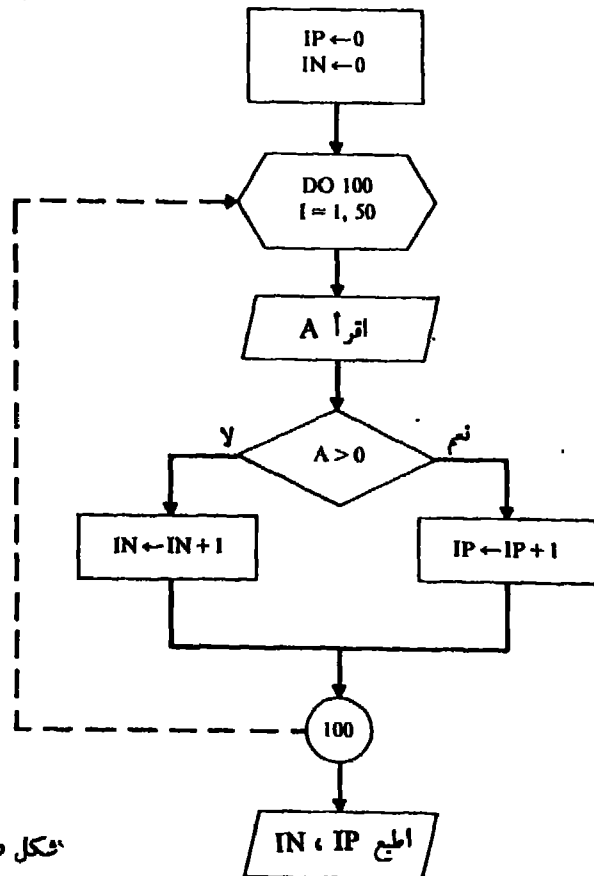


شكل ٥ - ٩

مطلوب نوع خاص من الجمل القابلة للتنفيذ يمكن للبدلين أن يشارا إليها لتتمكن الحلقة التكرارية DO من استكمال مسارها الطبيعي وقد خلقت جملة CONTINUE لهذا الغرض ومثال ذلك برنامج ب المصحح في قسم ٥ - ٧ .

```

IP = 0
IN = 0
DO 100 I = 1, 50
  READ(5, 10) A
  FORMAT(5, 10) A
  IF(A.GT.0.0) GO TO 50
  IN = IN + 1
  GO TO 100
50 CONTINUE
  WRITE(6, 20) IP, IN
  FORMAT(6X, 'POSITIVE = ', I3/
  1 6X, 'NEGATIVE = ', I3)
STOP
END
    
```



شكل ٥ - ١٠

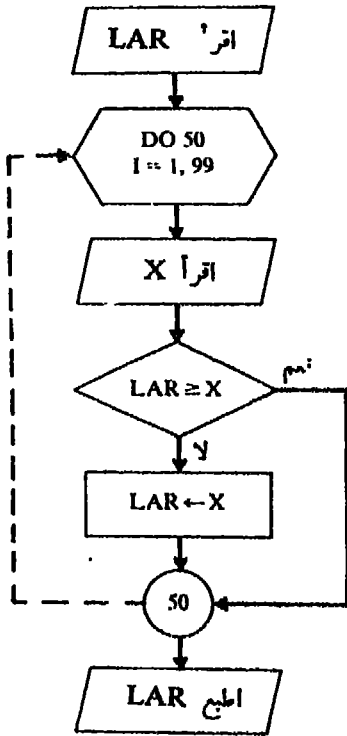
مثال ٥ - ٧

أعطيت مجموعة من 50 بطاقة . ثقب على كل بطاقة رقم غير الصفر . أكتب برنامج الفورتران الذي يمد عدد الأرقام الموجبة وعدد الأرقام السالبة .

تظهر خريطة سير العمليات للبرنامج وما يقابله بالفورتران في شكل د - ١٠ وقد اضطررنا هنا للمرة الثانية أن نستخدم جملة CONTINUE

مثال ٥ - ٨

اكتب برنامجاً لإيجاد أكبر رقم من 100 رقم ، حيث تم ثقب كل رقم على بطاقة منفصلة . تظهر خريطة سير العمليات وما يقابلها بالفورتران في شكل ه - ١١ . لاحظ أن القيمة النهائية بلزمة DO هي 99 وليس 100 وهذا نظراً لأن أول رقم يقرأ يخزن في LAR قبل تنفيذ حلقة DO التكرارية . مرة أخرى ، يجب أن نستخدم جملة CONTINUE



```

C
C PROGRAM TO FIND LARGEST OF 100 NUMBERS
C
REAL LAR
READ(5, 10) LAR
FORMAT(F10.2)
DO 50 I = 1, 99
    READ(5, 10) X
    IF(LAR.GE.X) GO TO 50
    LAR = X
50 CONTINUE
WRITE(6, 20) LAR
FORMAT(1X, F10.2)
STOP
END
  
```

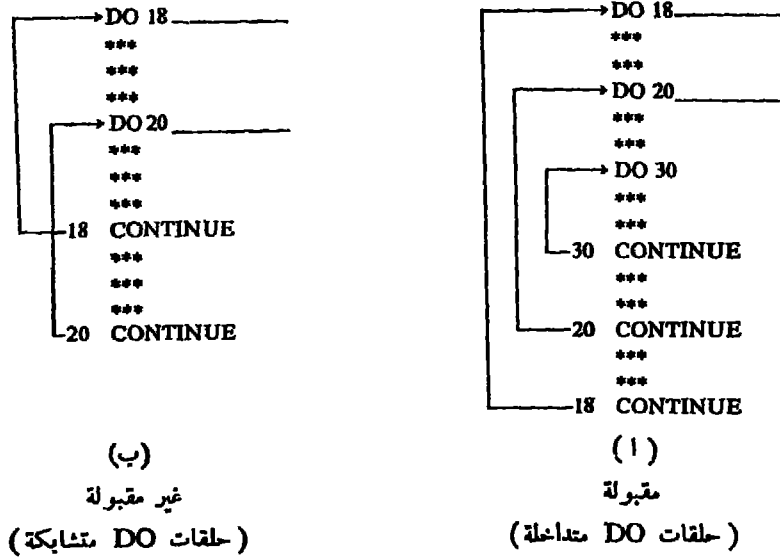
شكل ٥ - ١١

٥ - ٩ حلقات DO التكرارية المتداخلة

من الممكن أن تكون لدينا حلقة DO تكرارية (حلقة DO تكرارية داخلية) . تقع بالكامل داخل مدى حلقة DO تكرارية أخرى (حلقة DO تكرارية خارجية) . تسمى حلقات DO التكرارية التي تظهر بهذا النمط حلقات DO التكرارية المتداخلة والقوانين التي تطبق على حلقات DO التكرارية المتداخلة أساساً هي نفسها التي تطبق على حلقة DO التكرارية المفردة . بينما تراعى بعض النقاط الهامة التالية :

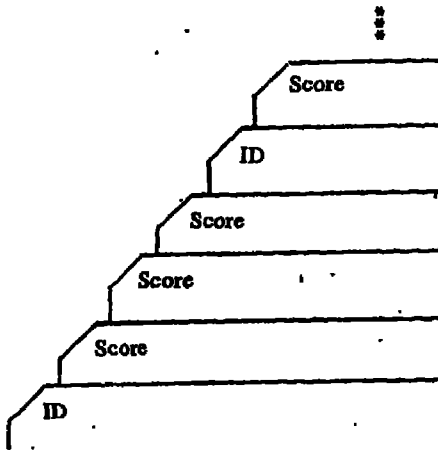
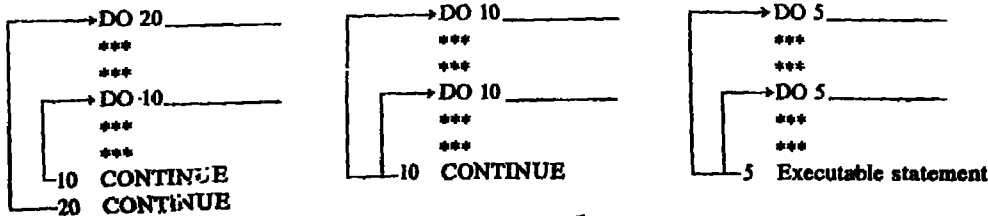
١ - حيث أن الدليل لا يمكن أن يعاد تعريفه داخل جسم حلقة DO التكرارية ، فدليل حلقة DO التكرارية الداخلية لا يجب أن يكون هو نفسه دليل حلقة DO التكرارية الخارجية .

٢ - يجب أن تقع حلقة DO التكرارية الداخلية تماماً بداخل حلقة DO التكرارية الخارجية ، أي ، لا يجب أن تتشابك الحلقات التكرارية . يوضح شكل ٥ - ١٢ طريقة تداخل حلقات DO التكرارية .



شكل ٥ - ١٢

٣ - تحدد عدد حلقات DO التكرارية المتداخلة بواسطة المترجم . يمكن أن يكون نقل التحكم بداخل حلقات DO التكرارية المتداخلة بصورة شاذة . ولكن القوانين تظل هي نفسها كما مع حلقات DO التكرارية المفردة . وأخيراً ، نلاحظ أن حلقات DO التكرارية المتداخلة يمكن أن تكون لها نفس الجملة الأخيرة . ويبين الشكل ٥ - ١٣ حلقات DO التكرارية المتداخلة المقبولة وهي جميعاً نفس الشيء .



مثال ٥ - ٩

أعطيت ثلاثة اختبارات لفصل دراسي به 25 طالباً وقد تم تقييم رقم ID للطلاب ودرجات الاختبار على بطاقات ورتبت كما في الشكل ٥ - ١٤ . اكتب برنامجاً لحساب متوسط الاختبارات لكل طالب .

يمكن حساب متوسط درجات الثلاث اختبارات بواسطة جزء البرنامج

التالي .

```

SUM = 0.0
DO 20 I = 1, 3
    READ(5, 10) SCORE
    FORMAT(F6.2)
    SUM = SUM + SCORE
20 CONTINUE
AVE = SUM/3.0

```

وحيث أنه يجب علينا أن نحسب المتوسط لكل من 25 طالب ، فيجب أن نكرر الجزء السابق 25 مرة . لذلك نحيط الجزء السابق بحلقة DO تكرارية أخرى كما يلي :

```

WRITE(6, 100)
100 FORMAT('1', 4X, 'ID', 9X, 'AVERAGE')
DO 80 K = 1, 25
    READ(5, 90) ID
    FORMAT(I10)
    SUM = 0.0
    DO 20 I = 1, 3
        READ(5, 10) SCORE
        FORMAT(F6.2)
        SUM = SUM + SCORE
    CONTINUE
    AVE = SUM/3.0
    WRITE(6, 30) ID, AVE
    FORMAT(1X, I10, 5X, F6.2)
80 CONTINUE

```

نفس الجزء السابق

ملاحظة : لاحظ زحزحة حلقة DO التكرارية الداخلية بداخل حلقة DO التكرارية الخارجية . ومرة أخرى ، يفضل هذا من أجل تسهيل القراءة مع العلم أن هذا لا يؤثر على تشغيل البرنامج .

مسائل محلولة

حلقات DO التكرارية :

٥ - ١ حدد عدد مرات تكرار كل حلقة من حلقات DO التكرارية وبين كذلك لأي قيمة من قيم الدليل سوف تنفذ كل حلقة .

(أ) DO 10 K = 1, 11, 3
(ب) DO 20 JIM = 4, 12
(ج) DO 30 LARGE = 8, 18, 15
(د) DO 40 M = 7, 4, 2

(أ) القيمة الابتدائية هي 1 ومعامل الزيادة هو 3 ومن ثم ستنفذ حلقة DO التكرارية طالما $K = 1 + 3 = 4$ و $K = 4 + 3 = 7$ و $K = 7 + 3 = 10$ وعندما تزداد K المرة التالية ستكون قيمتها أكبر من القيمة النهائية (11) لذا سينتقل التحكم خارج حلقة DO التكرارية . وبذلك ستنفذ حلقة DO التكرارية أربع مرات .

(ب) حيث أن قيمة معامل الزيادة غير موجود تكون قيمة معامل الزيادة 1 ومن ثم ستنفذ حلقة DO التكرارية طالما $JIM = 4, 5, \dots, 12$ وبذلك تنفذ تسع مرات .

(ج) تنفذ حلقة DO التكرارية مرة واحدة فقط وذلك عندما تكون $LARGE = 8$ ، وحيث أن القيمة التالية لـ LARGE هي $23 = 15 + 8$ وهي أكبر من قيمة الاختبار 18 فسوف يتوقف تنفيذ الحلقة التكرارية .

(د) رغم أن القيمة M الابتدائية أكبر من القيمة النهائية إلا أن حلقة DO التكرارية سوف تنفذ مرة واحدة عندما تكون $M = 7$ حيث أن الاختبار يتم في نهاية حلقة DO التكرارية (وستعطي بعض المترجمات رسالة خطأ) .

٢-٥ اكتشاف الأخطاء : إن وجدت ، في كل جملة من جمل DO :

DO 300 JILL = JOHN, JIM (ج) DO 100, K = 3, 24, 2 (أ)
DO 400 K = 4, 2*M, 3 (د) DO 200 JACK = 5, M, 4 (ب)

(أ) يجب ألا تكون هناك فصلة بعد 100 .

(ب) يجب ألا تكون هناك فصلة بعد 4 .

(ج) لا توجد هناك أخطاء (مع فرض أن JOHN و JIM قد تم تعريفهما)

(د) لا يمكن أن تتم أي عملية حسابية في أي من معاملات حلقة DO التكرارية .

يجب أن تتغير الجملة ، وليكن ، إلى الجملتين التاليتين .

MM = 2*M
DO 400 K = 4, MM, 3

٢-٥ أوجد القيمة النهائية لـ K بعد تنفيذ كل جزء من برنامج الفورتران :

(أ) K = 2
DO 10 I = 3, 8, 2
K = K + I
10 CONTINUE
K = 2*K
40 CONTINUE
K = 2*K

(ب) K = 2
DO 20 I = 3, 8, 2
K = K + I
IF(K.GT.6) GO TO 30
20 CONTINUE
30 K = 2*K
50 CONTINUE
K = 2*K

(أ) تخصص أول جملة 2 إلى K . ثم تنفذ بعد ذلك حلقة DO التكرارية كما يلي :

١- أولاً عندما I = 3 ينتج

$$K \leftarrow K + I = 2 + 3 = 5$$

٢- ثم عندما I = 5 ينتج

$$K \leftarrow K + I = 5 + 5 = 10$$

٣- ثم عندما I = 7 ينتج

$$K \leftarrow K + I = 10 + 7 = 17$$

تتجاوز قيمة I التالية قيمة الاختبار ، لذا ينتقل التحكم إلى الجملة الأخيرة التي تصاعف قيمة K

$$K \leftarrow 2*K = 2*17 = 34$$

وبذلك تكون قيمة K النهائية هي 34

(ب) تخصص الجملة الأولى 2 إلى K ثم تنفذ حلقة DO التكرارية كما يلي :

١- أولاً عندما I = 3 ينتج :

$$K \leftarrow K + I = 2 + 3 = 5$$

وحيث أن $K < 6$ فلا ينتقل التحكم إلى الجملة التي تحمل الرقم 30

٢- ثم عندما $I = 5$ ينتج

$$K \leftarrow K + I = 5 + 5 = 10$$

$$K \leftarrow K + I = 5 + 5 = 10$$

وحيث $K > 6$ فسوف ينتقل التحكم إلى الجملة التي تحمل الرقم 30 أي إلى الجملة الأخيرة .

تضاعف الجملة الأخيرة قيمة K لتصبح :

$$K \leftarrow 2 * K = 2 * 10 = 20$$

$$K \rightarrow 2 * K = 2 * 10 = 20$$

ومن ثم ، تكون قيمة K النهائية هي 20

(ج) حيث أن $M = 2$ وهذا هو نفس البرنامج كما في (أ) من ثم فقيمة K الأخيرة هي 34

(د) حيث أن $K = 2$ فجملة DO هي نفسها كما في (أ) ، ومع ذلك فإن K الآن معامل وقيمتها تغيرت في البرنامج ، وهذا غير مسموح به . وبذلك لا يمكن تنفيذ جزء البرنامج .

٥ - ٤ أو بد قيمة K النهائية بعد تنفيذ كل أجزاء برنامج الفورتران :

$K = 2$	(ب) $K = 2$	(١)
10 DO 20 I = 3, 8, 2	10 DO 20 I = 3, 8, 2	
IF(I.EQ.5) GO TO 10	IF(I.EQ.5) GO TO 20	
K = K + I	K = K + I	
20 CONTINUE	20 CONTINUE	
K = 2 * K	K = 2 * K	

(١) تخصص الجملة الأولى 2 إلى K ثم تنفذ حلقة DO التكرارية كما يلي :

(١) أولاً عندما $I = 3$ وحيث أن $I \neq 5$ تنفذ الجملة $K = K + I$ التي تعطي

$$K \leftarrow K + I = 2 + 3 = 5$$

(٢) ثم عندما $I = 5$. وحيث $I = 5$ لذا ينتقل التحكم إلى جملة CONTINUE التي تميد دورة الحلقة التكرارية .

(٢) ثم عندما $I = 7$ وحيث أن $I \neq 5$ تنفذ الجملة $K = K + I$ التي تعطي

$$K \leftarrow K + I = 5 + 7 = 12$$

تتجاوز قيمة I التالية قيمة الاختبار ، ولذا ينتقل التحكم إلى الجملة التالية لحلقة DO التكرارية ، التي تضاعف K . ومن ثم تكون قيمة K النهائية 24

(ب) تخصص الجملة الأولى 2 إلى K ثم تنفذ حلقة DO التكرارية كما يلي :

(١) أولاً عندما $I = 3$ وحيث أن $I \neq 5$ تنفذ الجملة $K = K + I$ التي تعطي

$$K = K + I = 2 + 3 = 5$$

(٢) ثم عندما $I = 5$ وحيث، أن $I = 5$ فلذا ينتقل التحكم إلى جملة DO . وحيث أن التحكم انتقل إلى جملة DO فتبدأ حلقة DO التكرارية من الأول مرة أخرى وتعمل $I = 3$ وبذلك تتكرر (١) و (٢) مرة ثانية وثالثة . ويعطى هذا حلقة تكرارية لا نهائية ، ولا توجد قيمة نهائية لـ K ، (اننا نؤكد أنه لو انتقل التحكم إلى جملة DO يعاد الدليل إلى قيمته الابتدائية) .

٥ - ٥ اكتشاف الأخطاء ، إن وجدت في كل برنامج . يمثل السهم المقوس نقل التحكم

<p>C THIRD PROGRAM</p> <p>.....</p> <p>DO 10 I = 1, 25</p> <p>.....</p> <p>DO 20 J = 1, 15</p> <p>.....</p> <p>20 CONTINUE</p> <p>.....</p> <p>10 CONTINUE</p> <p>.....</p> <p>END</p>	<p>(ب) C</p>	<p>FIRST PROGRAM (١)</p> <p>.....</p> <p>DO 10 I = 1, 25</p> <p>.....</p> <p>10 CONTINUE</p> <p>.....</p> <p>END</p>
<p>C FOURTH PROGRAM</p> <p>.....</p> <p>DO 10 I = 1, 25</p> <p>.....</p> <p>DO 20 J = 1, 15</p> <p>.....</p> <p>20 CONTINUE</p> <p>.....</p> <p>10 CONTINUE</p> <p>.....</p> <p>END</p>	<p>(د) C</p>	<p>SECOND PROGRAM (٢)</p> <p>.....</p> <p>DO 10 I = 1, 25</p> <p>.....</p> <p>DO 20 J = 1, 15</p> <p>.....</p> <p>10 CONTINUE</p> <p>.....</p> <p>20 CONTINUE</p> <p>.....</p> <p>END</p>

(١) لا يمكن نقل التحكم إلى منتصف حلقة DO تكرارية .

(ب) حلقات DO التكرارية متشابهة .

(ج) لا يمكن نقل التحكم من حلقة DO خارجية إلى داخل حلقة DO داخلية .

(د) لا توجد أخطاء . يمكن أن تنتقل من حلقة DO داخلية إلى حلقة DO خارجية

برامج :

٥ - ٦ اكتب البرنامج الذي يطبع الأعداد الصحيحة الموجبة من 1 إلى 300 مع طبع كل ثلاثة أرقام في سطر ، حتى يبدو المخرج كالشكل التالي :

```

1     2     3
4     5     6
.....
298   299   300

```

نريد طباعة I ، I + 1 ، I + 2 على كل سطر طالما I = 1, 4, 7, ... و I ≤ 300 . ومن ثم . نستخدم حلقة DO التكرارية بدليل I يتغير من 1 إلى 300 ومعامل الزيادة 3 ويبدو البرنامج كما يلي :

```

DO 100 I = 1, 300, 3
  J = I + 1
  K = I + 2
  WRITE(6, 10) I, J, K
10  FORMAT(1X, 3(1X, 3X))
100 CONTINUE
END

```

٥-٧ اكتب برنامج الفورتران الذي يقرأ عدداً صحيحاً فردياً موجباً N ويحسب (إلى ثلاثة أماكن عشرية) المجموع :

$$1 - 1/2 + 1/3 - \dots + 1/N \quad (\text{ج}) \qquad 1 + 1/2 + 1/3 + \dots + 1/N \quad (\text{أ})$$

$$1 + 1/3 + 1/5 + \dots + 1/N \quad (\text{ب})$$

(أ) أولاً اجعل $SUM = 0.0$ ثم استخدم حلقة DO تكرارية بدليل K حيث $K = 1, 2, \dots, N$ وذلك لجمع $1/K$ إلى SUM . وفيما يلي نبين البرنامج

```

READ(5, 10) N
10  FORMAT(I5)
SUM = 0.0
DO 100 K = 1, N
  X = FLOAT(K)
  SUM = SUM + 1.0/X
100 CONTINUE
WRITE(6, 20) N, SUM
20  FORMAT(1X, I5, 3X, F10.3)
STOP
END

```

(ب) البرنامج هو نفسه مثل (أ) فيما عدا أن حلقة DO التكرارية يجب أن تستبدل بالتالي

```
DO 100 K = 1, N, 2
```

حيث أننا سوف نستخدم قيم K فقط الفردية .

(ج) البرنامج هو نفسه مثل (أ) فيما عدا أنه يجب استبدال السطر السادس بما يلي

```
SUM = SUM + (-1.0)**(K + 1)/X
```

حيث أن الإشارات ستبدل .

٥-٨ افترض أنه تم إيداع مبلغ \$2000.00 في حساب توفير سنة 1977 وافترض أن البنك يدفع 6 في المائة فائدة مركبة سنوياً على الحساب . اكتب البرنامج الذي يطبع YEAR و AMOUNT للحساب حتى سنة 1995 (تقارن مع مسألة (٤ - ١١) .

تذكر أولاً أن كل سنة تزداد الـ AMOUNT بمقدار 6 في المائة .

```
AMOUNT ← AMOUNT + 0.06*AMOUNT
```

القانون العام هو

```
AMOUNT ← AMOUNT + RATE*AMOUNT = AMOUNT(1 + RATE)
```

حيث (RATE هو سعر الفائدة) . وفيما يلي نبين البرنامج

```

INTEGER YEAR
WRITE(6, 10)
10 FORMAT('1', 6X, 'YEAR', 6X, 'AMOUNT'//)
AMOUNT = 2000.00
YEAR = 1977
WRITE(6, 20) YEAR, AMOUNT
20 FORMAT(1X, I10, 3X, 'S', F8.2)
DO 99 YEAR = 1978, 1995
    AMOUNT = AMOUNT + 0.06*AMOUNT
    WRITE(6, 20) YEAR, AMOUNT
99 CONTINUE
STOP
END

```

(لاحظ أننا استخدمنا حلقة DO التكرارية بديل YEAR ابتداء من سنة 1978 حيث أن هذه هي أول سنة تخصص فيها الأرباح)

٥-٩ اعتبر أنه تم تجميع مجموعة من البيانات x_1, x_2, \dots, x_n على بطاقات ، رقم واحد في كل بطاقة ، وتحتوي المجموعة على بطاقة مقدمة (عنوان - بداية)
(١) اكتب جزءاً من برنامج لحساب المجاميع التالية :

$$SUM = x_1 + x_2 + \dots + x_n = \sum_{i=1}^n x_i$$

$$SUMSQ = x_1^2 + x_2^2 + \dots + x_n^2 = \sum_{i=1}^n x_i^2$$

(رمز التجميع Σ يناقش في قسم ٨-٩ (ج)) .

(ب) يعرف كل من المتوسط ، التباين والانحراف المعياري x_1, x_2, \dots, x_n بالآتي :

$$m = \text{mean} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\text{variance} = \frac{\sum_{i=1}^n (x_i - m)^2}{n}$$

$$\text{standard deviation} = \sqrt{\text{variance}}$$

مع ذلك فيمكن أن نكتب التباين كالآتي :

$$\text{variance} = \frac{\sum_{i=1}^n x_i^2}{n} - m^2$$

اكتب برنامجاً يحسب المتوسط ، التباين ، والانحراف المعياري للبيانات التي تم تجميعها .

(١) في البداية ، اجعل $SUM = 0.0$ و $SUMSQ = 0.0$. بعد قراءة N استخدم حلقة DO التكرارية لقراءة قيم X وكذلك لجمع X إلى SUM و X^2 إلى SUMSQ . وفيما يلي نبين جزء البرنامج :

```

SUM = 0.0
SUMSQ = 0.0
READ(5, 10) N
10  FORMAT(15)
   DO 100 I = 1, N
       READ(5, 20) X
       FORMAT(F10.2)
20      SUM = SUM + X
       SUMSQ = SUMSQ + X**2
100  CONTINUE

```

(ب) يمكننا القانون الثاني للتيبين من حساب التباين في نفس الوقت مع المتوسط (بدون استخدام المتغيرات ذات الدليل)
لذلك سوف نستخدم جزء البرنامج السابق في برنامجنا التالي :

```

REAL MEAN,
SUM = 0.0
SUMSQ = 0.0
READ(5, 10) N
10  FORMAT(15)
   DO 100 I = 1, N
       READ(5, 20) X
       FORMAT(F10.2)
20      SUM = SUM + X
       SUMSQ = SUMSQ + X**2
100  CONTINUE
   XN = FLOAT(N)
   MEAN = SUM/XN
   VAR = SUMSQ/XN - MEAN**2
   SD = SQRT(VAR)
30  WRITE(6, 30) MEAN, VAR, SD
   FORMAT(1X, 3(F10.2, 2X))
STOP
END

```

١٠-٥ طريقة مجموع الأرقام هي إحدى طرق حساب قيمة الاستهلاك . فعل سنيل المثال ، افترض أن سيارة سعرها \$4500.00 ستستهلك قيمتها على مدى خمس سنوات مجموع أرقام السنوات SUM في هذه الحالة هو

$$SUM = 1 + 2 + 3 + 4 + 5 = 15$$

إذا طبقنا الطريقة فإن 5/15 من \$4500 تستهلك في السنة الأولى ، 4/15 في السنة الثانية ، 3/15 في السنة الثالثة ، وهكذا ، يمكن أن نجد هذه البيانات كالتالي :

السنة	قيمة الاستهلاك
1	1500.00
2	1200.00
3	900.00
4	600.00
5	300.00

اكتب البرنامج الذي يتقبل عدداً موجباً حقيقياً $COST$ وعدداً صحيحاً موجباً N ويطبّق قيمة الاستهلاك $COST$ على مدى عدد N من السنوات باستخدام طريقة مجموع الأرقام حتى يبدو المخرج في الصورة السابقة .
سنتستخدم القانون :

$$SUM = 1 + 2 + 3 + \dots + n = n(n + 1)/2$$

لاحظ أن n/SUM من $COST$ تستهلك أول سنة ، $(n-1)/SUM$ السنة الثانية ، وهكذا ، وبالتحديد $(n-K+1)/SUM$ من $COST$ يستهلك في السنة K . نستخدم في البرنامج حلقة DO التكرارية بدليل $K = 1, 2, \dots, N$ ونحسب قيمة الاستهلاك باستخدام .

$$DEP = FLOAT(N - K + 1)/SUM * COST$$

وفيما يلي البرنامج :

```

C
C      PROGRAM USING SUM-OF-DIGITS DEPRECIATION
C
      READ(5, 10) COST, N
10    FORMAT(F15.2, I5)
      SUM = N*(N + 1)/2
      WRITE(6, 20)
20    FORMAT('1', 5X, 'YEAR', 5X, 'DEPRECIATION')
      DO 100 K = 1, N
          DEP = FLOAT(N - K + 1)/SUM * COST
          WRITE(6, 30) K, DEP
30    FORMAT(6X, I3, 6X, F10.2)
100   CONTINUE
      STOP
      END

```

١١ - ٥ (١) ادرس متعددة الحدود التريمية $y = 2x^2 - 3x - 5$ اكتب برنامج الفورتران الذي يجد y لقيم x والتي تراوح ما بين -4 و 4 بخطوات 0.5 .
(ب) ادرس متعددة الحدود .

$$z = x^3 - 3xy^2 + 2xy + y - 2y^3$$

اكتب برنامج فورتران باستخدام حلقة DO التكرارية المتداخلة التي تجد z لقيم x وتأخذ y القيم من -4 إلى 4 بخطوات 0.5 .

(١) حيث أن هناك 17 قيمة لـ x لذا سنستخدم حلقة DO التكرارية بدليل $I = 1, 2, \dots, 17$. العلاقة بين الدليل I و x هي كالتالي :

$$\begin{array}{l} x: \quad -4, \quad -3.5, \quad -3, \quad -2.5, \quad \dots, \quad 3.5, \quad 4 \\ I: \quad 1, \quad 2, \quad 3, \quad 4, \quad \dots, \quad 16, \quad 17 \end{array}$$

لاحظ أننا نستطيع أن نحصل على I من قيم x باستخدام

$$x = -4 + 0.5(I - 1)$$

وفيما يلي البرنامج :

```

C      PROGRAM QUADRATIC POLYNOMIAL
      DO 100 I = 1, 17
        X = -4.0 + 0.5*FLOAT(I - 1)
        Y = 2.0*X**2 - 3.0*X - 5.0
        WRITE(6, 10) X, Y
      10  FORMAT(1X, F10.3, 3X, F10.3)
      100 CONTINUE
      STOP
      END

```

(ب) يشابه هذا البرنامج ذلك الموجود في (أ) فيما عدا أننا الآن لدينا حلقة DO تكرارية لـ x وكذلك حلقة أخرى لـ y

```

C      PROGRAM POLYNOMIAL
      DO 200 I = 1, 17
        X = -4.0 + 0.5*FLOAT(I - 1)
        DO 100 J = 1, 17
          Y = -4.0 + 0.5*FLOAT(J - 1)
          Z = X**3 - 3.0*X*Y**2 + 2.0*X*Y + Y - 2.0*Y**3
          WRITE(6, 10) X, Y, Z
        100  FORMAT(1X, 3(3X, F10.2))
      100  CONTINUE
      200  CONTINUE
      STOP
      END

```

٥ - ١٢ الرسم البياني للمعادلة $x^2 + y^2 = 50$ هو دائرة C مركزها عند نقطة الأصل ونصف قطرها $\sqrt{50}$

(أ) حدد عدد النقط ذات الإحداثيات الصحيحة الموجبة التي تقع في الدائرة .

(ب) حدد عدد النقط ذات الإحداثيات الصحيحة التي تقع في الدائرة .

(أ) لاحظ أولاً أن قيم x و y لا يمكن أن تتجاوز 7 حيث أن نصف القطر هو $\sqrt{50}$. نستخدم حلقتين من حلقات DO التكرارية المتداخلة واحدة لـ x وواحدة لـ y . نبدأ أيضاً بعدد K باعطاءه قيمة ابتدائية 0 ولا نحسب تلك النقط (x, y) بحيث تكون $x^2 + y^2 \geq 50$. وفيما يلي ذلك البرنامج . لاحظ أن x و y قد تم تعريفها على أنهما متغيرات صحيحة .

```

C      PROGRAM COUNTING POINTS IN CIRCLE
      INTEGER X, Y
      K = 0
      DO 100 X = 1, 7
        DO 200 Y = 1, 7
          IF(X**2 + Y**2.GE.50) GO TO 200
          K = K + 1
        200  CONTINUE
      100  CONTINUE
      WRITE(6, 10) K
      10  FORMAT(1X, 'THE NUMBER OF POINTS IS', 2X, I5)
      STOP
      END

```

(ب) يجب أن نختبر هنا 15 قيمة لـ x و y أي 7, 6, ..., 1, 0, 1, 2, ..., 7 ومن ثم ، جهاز حلقة DO تكرارية بدليل $I = 1, 2, \dots, 15$ لـ x بحيث نجعل $x = 8 - I$ وحلقة DO تكرارية مثيلة لـ y وفيما عدا ذلك فالبرنامج يشابه الجزء (أ) .

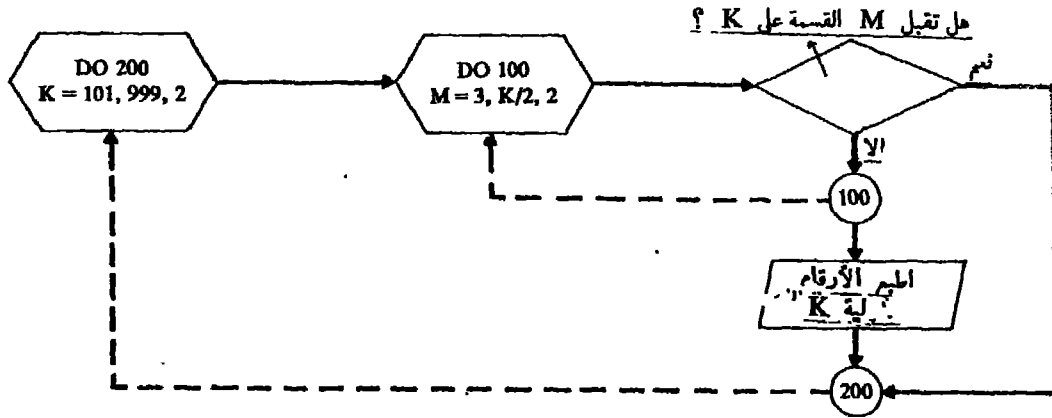
```

INTEGER X, Y
K = 0
DO 100 I = 1, 15
  X = 8 - I
  DO 200 J = 1, 15
    Y = 8 - J
    IF(X**2 + Y**2.GE.50) GO TO 200
    K = K + 1
200  CONTINUE
100  CONTINUE
WRITE(6, 10) K
10  FORMAT(1X, 'THE NUMBER OF POINTS IS', 2X, I5)
STOP
END

```

١٣ - ٥ أوجد كل الأرقام الأولية ذات الخانات الثلاث ، أي أوجد كل الأرقام الأولية ما بين 100 و 999 (قارن مع المثال ٦ - ٥) .

لا يمكن أن يكون أحد هذه الأرقام الأولية زوجياً لذلك فهنا نعتبر الأعداد الصحيحة الفردية لـ K فقط بين 101 ، 999 نرى ما إذا كانت K أولية . بالإضافة إلى ذلك ، إذا كانت K غير أولية ، فيجب أن يكون K قاسم فردى M بين 3 و $K/2$ وبذلك ، نستخدم حلقة DO تكرارية بدليل M لتوليد القواسم المحتملة (الممكنة) لـ K المعطاة . تظهر خريطة سير العمليات وترجمتها إلى الفورتران في شكل ١٥ - ٥ (أ) و (ب) ، على الترتيب .



شكل ١٥ - ٥ (أ)

```

C
C PROGRAM PRINTING PRIME NUMBERS
C
DO 200 K = 101, 999, 2
  KK = K/2
  DO 100 M = 3, KK, 2
    IF(K.EQ.(K/M)*M) GO TO 200
100  CONTINUE
WRITE(6, 10) K
10  FORMAT(1X, I10)
200 CONTINUE
STOP
END

```

شكل ١٥ - ٥ (ب)

مسائل تكميلية

حلقات DO التكرارية .

١٤ - ٥ حدد عدد مرات تنفيذ حلقة DO التكرارية ، وقم الدليل التي تنفذ بها الحلقة التكرارية إذا كانت جملتها الأولى هي :

DO 300 I = 6, 9, 5 (ج) DO 100 L = 2, 15, 3 (١)
DO 400 J = 8, 5, 4 (د) DO 200 JOHN = 5. 11 (ب)

١٥ - ٥ اكتشف الأخطاء ، إن وجدت في كل جملة من جمل DO التكرارية .

DO 900 K = I, J, K, (ج) DO 700, LAMB = 1, 14, L, (١)
DO 1000 LONG = K234, K123, K345 (د) DO 800 J = 7, M**2, 2 (ب)

١٦ - ٥ أوجد قيمة K بعد تنفيذ كل جزء من برامج الفورتران التالية :

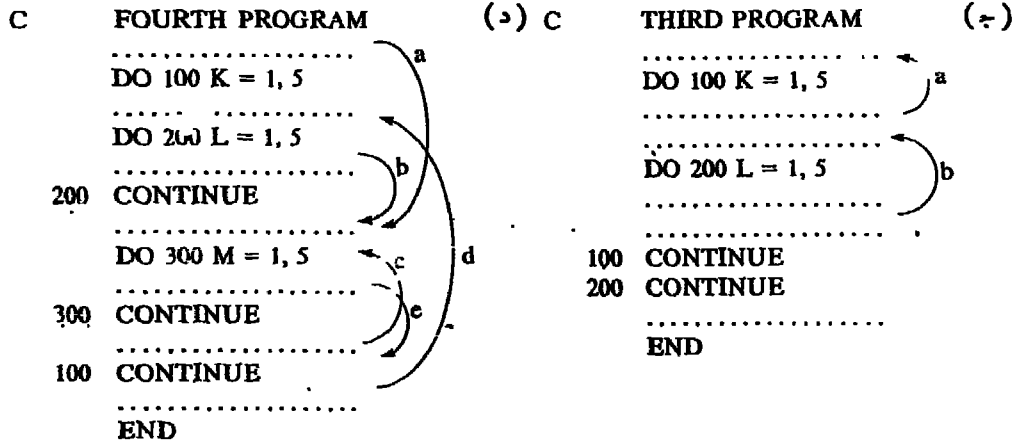
<p>K = 3 M = 2 DO 300 J = 3, 7, M M = M + K 300 CONTINUE K = 3*K</p> <p>K = 3 M = 2 DO 400 J = M, 7, M K = J + K + M IF(K.GT.9) GO TO 10 400 CONTINUE 10 K = 3*K</p>	<p>(ج)</p> <p>(د)</p>	<p>K = 3 DO 100 J = 3, 7, 3 (١) K = K + J 100 CONTINUE K = 3*K</p> <p>(ب)</p> <p>K = 3 M = 2 DO 200 J = 3, 7 M K = K + J 200 CONTINUE K = 3*K</p>
--	-----------------------	---

١٧ - ٥ أوجد قيمة K النهائية بعد تنفيذ كل جزء من برامج الفورتران التالية :

<p>K = 3 M = 2 20 DO 10 J = 4, 9, M IF(J.EQ.6) GO TO 20 K = K + J**2 10 CONTINUE K = 3*K</p>	<p>(ب)</p>	<p>(١)</p> <p>K = 3 M = 2 20 DO 10 J = 4, 9, M IF(J.EQ.6) GO TO 10 K = K + 2*J 10 CONTINUE K = 3*K</p>
--	------------	--

١٨ - ٥ افترض أن كل سهم مقوس يمثل انتقالا في التحكم . أوجد الأخطاء في كل برنامج من البرامج التالية :

<p>G SECOND PROGRAM (ج) C</p> <p>..... DO 100 K = 1, 5 DO 100 L = 1, 5 100 CONTINUE END</p>	<p>FIRST PROGRAM (١)</p> <p>..... DO 100 K = 1, 5 100 CONTINUE END</p>
---	--



برامج

(كثير من المسائل هي نفسها كالمسائل الموجودة في الفصل الرابع . بينما يجب أن تحل الآن استخدام حلقة DO فضلا عن استخدام عداد) .

١٩ - ٥ اكتب البرنامج الذي يطبع كل رقم فردي ذو خانتين N ومربعة N^2 وتكبيية N^3 حيث تظهر قيم N المختلفة على أسطر مختلفة (قارن مع المسألة ٤ - ٩)

٢٠ - ٥ اكتب جزءاً من برنامج فورتران يطبع الرقم 20 عشرين مرة ، الرقم 19 تسع عشرة مرة ، الرقم 18 ثمان عشرة مرة ، وهكذا

٢١ - ٥ افرض عددين صحيحين موجبين N و K في الذاكرة حيث $K < N$. اكتب جزء البرنامج مستخدماً جملة DO حيث تطبع قيمة N ثم تطبع قيمة $N-1$ ثم تطبع قيمة $N-2$ وهكذا إلى أن تصل إلى K ونطبع قيمتها . (تلميح : انظر مثال ٥ - ٥ (ب)) .

٢٢ - ٥ إذا أعطيت قيم A ، B ، N . اكتب جزء من برنامج فورتران لحسب

$$\frac{1}{A} + \frac{1}{A+B} + \frac{1}{A+2B} + \frac{1}{A+3B} + \dots + \frac{1}{A+NB}$$

٢٣ - ٥ اكتب البرنامج الذي يقرأ عدداً صحيحاً موجباً $N \geq 10$ ثم يحسب حاصل الضرب التالي إلى خمسة أماكن عشرية .

$$\frac{1}{1^2} \cdot \frac{3}{2^2} \cdot \frac{5}{3^2} \cdot \dots \cdot \frac{2N-1}{N^2}$$

٢٤ - ٥ اكتب البرنامج الذي يطبع كل الأعداد الصحيحة الموجبة الفردية الأقل من 100 مع حلف تلك الأعداد الصحيحة التي تقبل القسمة على 7 :

$$1, 3, 5, 9, 11, 15, 17, 19, 23, \dots, 97, 99$$

٢٥ - ٥ افرض ان AMOUNT هي قيمة وديعة فائدتها RATE (مركبة سنوياً) ، والرقم N عدد السنوات كلها تقرأ كميان : اكتب البرنامج الذي يطبع مقدار (AMOUNT) المبلغ كل ستة لمدة قدرها N من السنوات (قارن مع مسألة ٥ - ٨) .

١١ - البرمجة بلغة الفورتران

- ٢٦ - ٥ افرض أننا أودعنا \$ 500.00 كل سنة في حساب توفير يعطى 7 في المائة فائدة مركبة سنوياً . اكتب البرنامج الذى يطبع المبلغ الموجود في الحساب كل سنة ولمدة 10 سنوات .
- ٢٧ - ٥ افرض أننا أودعنا \$ 500.00 كل سنتين في حساب توفير يعطى 7 في المائة فائدة مركبة سنوياً . اكتب البرنامج الذى يطبع المبلغ الموجود في الحساب كل سنة ولمدة 10 سنوات .
- ٢٨ - ٥ افترض رجل مبلغ \$ 300.00 من بنك بمعدل فائدة 1.5 في المائة شهرياً ، وهو يسد \$ 25.00 عند نهاية كل شهر . وبذلك ، فمعد نهاية الشهر الأول يكون مدان بالمبلغ التالي :

$$\text{AMOUNT} + \text{INTEREST} - \text{REPAYMENT} = 300 + (1.5\%)(300) - 25 \\ = 300 + 4.50 - 25 = 279.50$$

- (!) اكتب برنامج الفورتران الذى يطبع المبلغ المدان به كل شهر ولمدة سنة .
- (ب) اكتب جزء من برنامج فورتران يحسب عدد الشهور التى يجب أن يدفع بها أقساط هذا القرض ، وقيمة آخر قسط .
- ٢٩ - ٥ أوجد عدد النقط ذات الاحداثيات الصحيحة التى تقع داخل القطع الناقص الممثل بالمعادلة $2x^2 + 3y^2 = 100$
- ٣٠ - ٥ ثلاثة أعداد صحيحة موجبة a و b و c حيث $a < b < c$ تشكل ثلاثية فيثاغورث إذا كانت $a^2 + b^2 = c^2$. عل سبيل المثال 3 ، 4 ، 5 تشكل ثلاثية فيثاغورث حيث أن $3^2 + 4^2 = 5^2$. اكتب البرنامج الذى يجد كل ثلاثيات فيثاغورث a, b, c حيث $a, b < 25$.
- ٣١ - ٥ افرض أن كل بطاقة في مجموعة بطاقات تحتوي على رقم حقيقى . أضيفت بطاقة مقدمة تحتوي على الرقم N وهو عدد البطاقات في المجموعة . أوجد أصغر وأكبر قيمة .
- ٣٢ - ٥ افرض أنه تم تفتيح 25 عدد صحيح موجب على بطاقات ، عدداً في كل بطاقة . اكتب البرنامج الذى يجد العدد التالي لأكبر عدد صحيح .
- ٣٣ - ٥ افرض مجموعة من البطاقات تحتوي على عدد واحد صحيح موجب لكل بطاقة . اكتب البرنامج الذى يجد أكبر عدد صحيح زوجي في المجموعة ، أو يطبع NO EVEN INTEGER إذا كانت كل الأعداد الصحيحة فردية . اعتبر المجموعة لها بطاقة مقدمة تبين عدد البطاقات N في المجموعة .
- ٣٤ - ٥ افرض مجموعة بطاقات تحتوي على عدد واحد صحيح في كل بطاقة . اكتب البرنامج الذى يجد عدد الأرقام الصحيحة الزوجية ، وعدد الأرقام الصحيحة الفردية (اعتبر المجموعة لها بطاقة مقدمة) .
- ٣٥ - ٥ الثابت π يمكن أن يقرب بواسطة الصيغة الرياضية التالية :

$$\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots$$

اكتب البرنامج الذى

(أ) يجمع أول 1000 حد

(ب) يجمع أول 1000 حد في الترتيب العكسى .

(سؤال : هل هناك أى فرق ؟)

٢٦-٥ اعتبر المعادلة $y = x^2 - 4x + 6$. اكتب برنامج الفورتران الذى يجد y لقيم x :

(أ) x من 5 إلى 5 بمعامل زيادة 0.1

(ب) من I إلى J بمعامل زيادة $D = 1/N$ ، حيث I, J, N متقبة على بطاقة و $I < J$.

يجب أن تطبع كل قيمة لـ x وقيمة y المقابلة (لما على سطر مختلف)

٢٧-٥ اعتبر المعادلة $z = x^2 - 2xy + 3y^2 - 8x + 3y - 8$. اكتب البرنامج الذى يجد قيمة z لقيم x و y حيث x و y تتراوح ما بين -3 و 3 بمعامل زيادة 0.2

٢٨-٥ عدل المسألة السابقة ٢٧-٥ بحيث يحدد البرنامج أيضاً أقصى قيمة لـ z

٢٩-٥ يمكن أن نحسب قيمة تقريبية لـ جيب x بالتقدير الدائرى وذلك بتجميع أول N حد من المتسلسلة التالية :

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

(أ) اكتب البرنامج الذى يحسب الجيب لـ 3 درجات بالتقدير الدائرى باستخدام المتسلسل السابقة و $N = 1, 2, \dots, 8$

تارن كل إجابة بتلك التى نحصل عليها باستخدام الدالة المكتوبة SINX

(ب) اكتب البرنامج الذى يقبل عدداً صحيحاً موجباً N وقيمة حقيقية x ويحسب قيمة جيب x باستخدام المتسلسلة السابقة .

(ج) اكتب البرنامج الذى يقبل قيمة حقيقية x ويحسب قيمة x بتجميع حدود متتالية من المتسلسلة السابقة إلى أن تكون

قيمة الحد المطلقة أقل من 10^{-15} .

اجابات للمسائل التكميلية المختارة

١٤-٥ (أ) خمس مرات 2 ، 5 ، 8 ، 11 ، 14 (ب) سبع مرات 5 ، 6 ، 7 ، 8 ، 9 ، 10 ، 11

(ج) مرة واحدة 6 (د) مرة واحدة 8

١٥-٥ (أ) لا يجب أن تكون هناك فصلة بعد 700 أو بعد L

(ب) لا يمكن أن نستعمل $M \neq 2$ كعامل

(ج) لا يجب أن تكون هناك فصلة بعد K . متغير الدليل K لا يمكن أن يكون واحداً من المعاملات .

(د) لا توجد أخطاء .

١٦-٥ (أ) 36 (ب) 54 (ج) خطأ لأن المعامل M تم تغييره (د) 39

١٧-٥ (أ) 81 (ب) الحاسب لن ينتهى من هذا الجزء من البرنامج (انظر المسألة المحلولة ٥ - ٤ (ب))

١٨-٥ (أ) النقل غير مسموح به .

(ب) تنقلات a ، c غير مسموح بها .

(ج) يجب أن تبديل جمل CONTINUE

(د) تنقلات a ، d غير مسموح بها .

الفصل السادس

المجموعات المتراسة والمتغيرات ذات الأداة

٦ - ١ مقدمة

إن المتغيرات التي استخدمناها إلى الآن تسمى أيضاً متغيرات بدون أداة أو متغيرات غير متجهة ، وكل متغير من هذه المتغيرات غير المتجهة يمثل خلية ذاكرة يمكن تخزين قيمة واحدة بداخلها ، فثلاً ، يمكن اعتبار متغيراً باسم X كصندوق :

X	
---	--

أحياناً كثيرة يمكن أن نستخدم نفس الإسم للإشارة إلى قائمة من القيم لما نفس المصائص المشتركة ، مثلاً أرقام البطاقات الشخصية لطلبة في فصل عام الحاسبات ودرجات الاختبار النهائي ، وهكذا ويمكن إنجاز ذلك باستخدام المجموعات المتراسة أو المتغيرات ذات الأداة. في الرياضيات ، يستخدم رمز بأداة مختلفة للإشارة إلى العناصر في مجموعة مطعاة على سبيل المثال S_1 ، S_2 ، ... ، S_{25} يمكن أن تشير إلى درجات 25 طالب في فصل . وباستخدام الترميز بالأداة يمكن حينئذ كتابة :

$$\sum_{i=1}^{25} s_i = s_1 + s_2 + \dots + s_{25}$$

وذلك لتجميع الدرجات ، و

$$\left(\sum_{i=1}^{25} s_i \right) / 25$$

لإيجاد متوسط درجات الفصل ، بمعنى آخر فاستخدام الأداة ضروري جداً في تطوير ترميز (مختصر) للأرقام المشمولة عند استخدام الجبر .

وفي الفورتران تستخدم الأداة أيضاً ، ولكن لها مظهر مختلف . بمعنى أن كل جملة فورتران يجب أن تكتب (أو تكتب) على سطر واحد ، فنستخدم الأقواس لتحيط الأداة ، فثلاً يمكن أن نكتب في الفورتران $S(1)$ ، $S(2)$ ، ... ، $S(25)$ بدلا من S_1 ، S_2 ، ... ، S_{25} . تلك المجموعات والتي تعرف عناصرها بدليل واحد تسمى في الفورتران مجموعات متراسة خطية أو ذات بعد واحد ، المجموعات المتراسة المتشابهة التي تعرف عناصرها بدليلين أو أكثر تسمى مجموعات متراسة ذات أبعاد متعددة .

٦ - ٢ مجموعات متراسة ذات بعد واحد

المجموعات المتراسة الخطية أو ذات البعد الواحد ، والتي بها N عنصر هي قائمة أو سلسلة من عدد N من خلايا الذاكرة ، حيث يمكن أن تخزن قيمة واحدة بكل خلية . هذه الخلايا لها نفس الإسم ومجاورة بعضها لبعض . فنعرف على أي خلية محددة في هذه المجموعة المتراسة ذات بعد الواحد بواسطة متغير صحيح K ، يرمز إلى مكانها في المجموعة المتراسة . (وبذلك يجب أن تكون قيمة $1 \leq K \leq N$) .

على سبيل المثال افرض أن ID هي مجموعة متراسة ذات بعد واحد بها 50 عنصر ، فيمكن أن ننص: ID صف من 50 صندوق كما في الشكل ٦ - ١ . بشار إلى كل خلية في المجموعة المتراسة ID باسمها ومكانها ، ويسمى دليل سفل . بالتحديد تشير ID (1) إلى العنصر الأول من المجموعة المتراسة ، وتشير ID(2) إلى العنصر الثاني ، وهكذا . وعلاوة على ذلك إذا عرفت K وكانت $1 \leq K \leq 50$ فإن ID(K) تشير إلى العنصر رقم K من المجموعة المتراسة

ID	
ID	
ID	

دليل

← 1

← 2

← 3

⋮

ID	
ID	
ID	

← 48

← 49

← 50

شكل ٦ - ١

تطبق القواعد المستعملة في تسمية المتغيرات العادية أيضاً على المجموعات المتراسة ، أي ، يتكون اسم المجموعة المتراسة من واحد إلى ستة حروف أبجدية رقية يجب أن يكون الأول منها أبجدياً . إذا كان أيضاً اسم المجموعة المتراسة يبدأ بالحرف I أو J أو K أو L أو N فيمكن أن تحتوي المجموعة المتراسة على أعداد صحيحة فقط... بالمثل فالمجموعات المتراسة التي تبدأ بحروف أخرى يمكن أن تحتوي على أعداد حقيقية ، رغم أنه يمكن التغلب على التسمية التقليدية بجملة نوع (والتي تناقش في قسم ٦ - ٣) ، ولكن يجب أن تكون عناصر المجموعة المتراسة دائماً من نفس النوع .

رغم أن أي عنصر من عناصر المجموعة المتراسة يحدد باسم المجموعة المتراسة متبوعاً بالدليل المناسب (محاط بأقواس) ، فيمكن أن يظهر كل عنصر في المجموعة المتراسة في التسميات الحسابية مثل المتغيرات غير المتجهة تماماً . على سبيل المثال التسميات التالية صحيحة لنوعياً :

$$\begin{aligned} X &= \text{AMOUNT}(5) + Q \\ \text{PAY}(K) &= \text{HOUR}(K) * 40.0 \\ \text{TAX}(6) &= 25.75 \end{aligned}$$

ومن ناحية أخرى ، لا يمكن أن نستخدم أسماء المجموعات المتراسة نفسها في الجمل الحسابية - على سبيل المثال ، إذا كانت GRADE هي مجموعة متراسة و SUM هي متغير عادي فالجمل التالية مثلاً :

$$\begin{aligned} \text{SUM} &= \text{GRADE} + \text{SUM} \\ \text{GRADE} &= \text{SUM} + 10.0 \\ \text{GRADE} &= \text{GRADE} + 20.0 \end{aligned}$$

غير مسموح بها . وهي لأن الإسم GRADE يشير إلى المجموعة المتراسة بأكملها بينما الحسابات تشمل على أماكن ذاكرة فردية .

يمكن أيضاً أن تشير إلى عناصر المجموعة المتراسة في جمل الإدخال / الإخراج على سبيل المثال :

READ(5, 10) X, TAX(3)

تأمر الحاسب أن يقرأ رقمين . يخزن الرقم الأول في المكان X ويخزن الثاني في العنصر الثالث من المجموعة المتراسة المسماة TAX ونستطيع أن نطلع من الذاكرة أرقام تحقيق الشخصية ID(1) ، ID(2) ، ... ، ID(K) كل رقم في سطر باستخدام جزء البرنامج التالي :

```
DO 100 I = 1, K
      WRITE(6, 20) ID(I)
20   FORMAT(11X, I10)
100 CONTINUE
```

رغم أن أسماء المجموعات المتراسة لا يمكن أن تظهر في الجمل الحسابية ، لكنها يمكن أن تظهر في جمل الإدخال / الإخراج وسيناقش هذا فيما بعد .

ملحوظة : يستخدم المصطلح متغيرات ذات أدلة في كتب مختلفة ليعني أشياء مختلفة . فبعض الكتب تستخدم متغيرات ذات أدلة ليعني المجموعة المترابطة نفسها . في حين أن كتباً أخرى تستخدم متغيراً ذا دليل ليعني عنصراً في المجموعة المترابطة وكتب أخرى تقول أن العناصر (24) ID ، (3) ID... الخ ، هي متغيرات ذات أدلة. لا تستخدم كثير من كتيبات IBM في لغة الفورتران المصطلح متغيراً ذا دليل على الإطلاق . ولكن كل الكتب تستخدم المصطلح متغيراً بدون دليل ليعني نفس الشيء وهي المتغيرات العادية غير المتجهة .

٦ - ٣ جملة DIMENSION

قبل أن تستخدم أى مجموعة مترابطة خطية أو متعددة الأبعاد في برنامج . يجب أن نمد المترجم بالمعلومات الآتية :

- ١ - اسم المجموعة المترابطة
- ٢ - عدد الأدلة في المجموعة المترابطة (أى ، ما إذا كانت المجموعة المترابطة ذات بعد واحد أو ذات بعدين ، الخ) .
- ٣ - العدد الكلي لأماكن الذاكرة التي ستخصص لهذا الإسم أو بتحديد أكثر القيمة المعطى لكل دليل .

يتم هذا باستخدام جملة النوع DIMENSION (وكما هو الحال في كل جملة النوع ، فإن جملة DIMENSION جملة غير قابلة للتنفيذ ويجب أن توضع قبل أى جملة قابلة للتنفيذ في البرنامج) .
وفيما يلي جملة DIMENSION النموذجية :

DIMENSION AMOUNT(100)

تدل هذه الجملة نترجم على أن AMOUNT هو اسم مجموعة مترابطة خطية حقيقية يجب أن تخصص لها عدد من أماكن الذاكرة مقدارها 100 بدلا من ذكر المجموعات المترابطة في جملة منفصلة فيمكن أن تعرف عدة مجموعات مترابطة في جملة DIMENSION واحدة. وعلى سبيل المثال :

DIMENSION ID(50), TAX(150)

تعرف ID على أنها مجموعة مترابطة خطية صحيحة بها 50 عنصراً و TAX مجموعة مترابطة خطية حقيقية بها 150 عنصراً . (شكل المجموعات المترابطة ذات الأبعاد المتعددة متشابهة وستناقش فيما بعد) . ترتيب ذكر المجموعات المترابطة في جملة DIM ليس مهما ، طالما أن المجموعات المترابطة كلها متضمنة في جملة DIMENSION ، لاحظ أيضاً أنه ليس هناك فصلة بعد كلمة DIMENSION . ولكن المجموعات المترابطة المختلفة تفصل فيما بينها بفصلات في جملة DIMENSION .

عموماً الدليل الصفري أو السالب غير مسموح به . ومن الواضح ، يجب ألا نستعمل دليلاً أكبر من الحجم الأقصى المحدد في جملة DIMENSION . على سبيل المثال ، إذا أعطينا جملة DIMENSION السابقة ، يجب ألا نستخدم ID(55) أو TAX(182) في البرنامج . يجب توخي الحرص بصفة خاصة من هذه الناحية ، حيث أن مترجمات كثيرة لا تتأكد من صحة الأدلة . وبالتالي يمكن ألاكتشف أخطاء من هذا النوع ولكنها ستنفذها مما يؤدي إلى نتائج مليئة بالأخطاء .

ذكرنا أن جملة DIMENSION تخصص العدد الأقصى من أماكن الذاكرة المطلوبة في مجموعة مترابطة ، رغم أن مخطط البرنامج يمكن أن يستخدم جزءاً من خلايا المجموعة المترابطة فقط . عموماً هذا التخصيص الزائد قد يكون ضرورياً ، ولكن يجب أن يتم حسب المسألة نفسها ومتطلباتها . على سبيل المثال ، يمكن ألا يعرف العدد الحقيقي للطلبة الذين أدوا الاختبار مثلا ، بعض الطلبة قد يتنبهون ، ولكن يمكن استخدام

DIMENSION TEST(60)

أغذين في الاعتبار أن الحجم الطبيعي لفصل دراسي لن يسع أكثر من 60 . وثوكد أن استهلاك مجموعات مترابطة كبيرة بدرجة غير مقبولة سيكون مكلفاً بالنسبة إلى وقت التشغيل والبيئة الحاسوبية ككل .

ملاحظة : وأتأ فؤكذ أن جملة DIMENSION تخصص خلايا الذاكرة للمجموعات المتراسة أثناء عملية الترجمة للبرنامج ؛ وليس أثناء التنفيذ ، من ثم فالذييل المذكور في أى جملة DIMENSION يجب أن يكون ثابتاً صحيحاً بدون إشارة (الاستثناء الوحيد المبرحيد قد يحدو في البرامج الفرعية SUBPROGRAMS أنظر الفصل السابع) وبالتالي فإن الجبل في الشكل .

DIMENSION AMOUNT(K)

N = 75
DIMENSION SCORE(N)
REAL(5, 10) I
DIMENSION GRADE(I)

أو

غير مقبولة .

هناك نقطة أخيرة نريد أن نذكرها في هذا القسم . إفرض أننا نريد SCORE كإسم لمجموعة متراسة خطية تحتوي على 50 عدداً صحيحاً على الأكثر (وليست أرقاماً حقيقية) يمكن عمل هذا باستعمال جمل النوع :

INTEGER SCORE
DIMENSION SCORE(50)

إلا أنه ، يمكن أن نستخدم جملة واحدة كبديل وهي :

INTEGER SCORE(50)

حيث أنها تعطي أيضاً المعلومات الهامة إلى المترجم . بالمثل :

REAL INVEST(150)

سعر INVEST كإسم للمجموعة المتراسة الخطية التي بها 150 قيمة حقيقية . وكالمعاد يمكن أن يظهر أكثر من بند من المعلومات في : $n \times m$ ، هذه ، ويجب فصل البنود عن بعضها بواسطة فصالات .

٦ .٤ التعبيرات الرياضية للأدلة

إحدى المزايا الرئيسية لاستخدام المجموعات المتراسة هي أن الأدلة نفسها يمكن أن تكون متغيرات . في الحقيقة يمكن أن تكون تميرات حسابية مقصورة على أحد الأشكال الآتية :

$$n, \quad K, \quad K + n, \quad n * K, \quad n * K \pm m$$

حيث n ، m ثوابت صحيحة و K أى متغير صحيح . على سبيل المثال ، مايل يعتبر صحيحاً لنوعياً :

$$A(8), \quad A(K), \quad S(K - 2), \quad A(3 * K), \quad A(4 * K - 3)$$

ولكن التعبيرات الآتية لا تكون مقبولة عادة .

$$A(-4) \text{ (الدليل السالب غير مسموح به)}$$

$$A(0) \text{ (الصفير غير مسموح به كدليل)}$$

$$A(K * 2) \text{ (يجب أن تكتب } A(2 * K) \text{)}$$

$$A(K ** 2) \text{ (الدليل ليس أحد الأشكال السابقة)}$$

$$A(.TRUE.) \text{ (لا تعتبر الأدلة قيماً منطقية (أنظر قسم ٩ - ٦))}$$

$$A(3 * M(2)) \text{ (متغيرات غير متجهة هي التي يمكن أن تظهر فقط في تميرات الدليل)}$$

اننا نذكر أن بعض الحاسبات الكبيرة الحديثة تسمح بأى تعبيرات حسابية صحيحة القيمة كأدلة ، ويجب أن تعمل القواعد السابقة تبعاً لذلك . مع أننا إلى الآن قد ناقشنا فقط مجموعات متراسة ذات بعد واحد إلا أن القواعد السابقة على الأدلة تطبق أيضاً على المجموعات المتراسة متعددة الأبعاد .

٥ - أمثلة لاستخدام المجموعات المتراسة

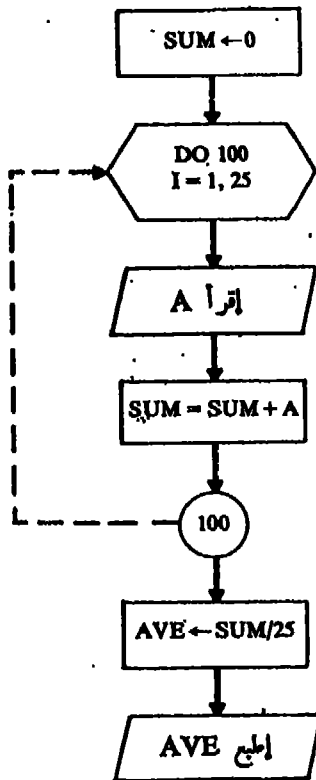
افرض أنه تم عقد امتحان لفصل دراسى أقل من 25 طالب ، وثقبت الدرجات على بطاقات ، درجة واحدة و كل بطاقة . والمطلوب كتابة برنامج فورتران لعمل الآتى :

١ - إيجاد متوسط الفصل الدراسى

٢ - إيجاد عدد الطلبة الذين كانت درجاتهم أقل من المتوسط .

أولا سنناقش الحالة الأبسط وهى أن هناك فعلا 25 طالباً أدوا الامتحان (لإيجاد متوسط الفصل الدراسى ، يجب أن نجد أولاً المجموع الكلى لكل الدرجات ويستلزم ذلك أن نتذكر المناقشة فى قسم ٤ - ١٠ لإيجاد مجموع عدة أرقام) .

طريقة وحيدة لفروض فى هذه المسألة هى باستخدام 25 إسماً متتيراً أو ليكن A_1, A_2, \dots, A_{25} وذلك لعدد 25 درجة . وهذا يؤدي إلى جمل طويلة جداً تكون معرضة للخطأ . (انظر صفحة ١٤٤) بالإضافة إلى أننا سنحتاج 25 جملة IF حتى يقارن كل درجة بمتوسط الفصل . علاوة على ذلك ، يكون البرنامج غير قابل للسيطرة إذا تناولنا على سبيل المثال 100 رقم أو أكثر . لذلك يجب أن نبحث عن طريقة أخرى لحل المسألة .



```

SUM = 0.0
DO 100 I = 1, 25
    READ(5, 10) A
    10    FORMAT(F10.1)
    SUM = SUM + A
100 CONTINUE
AVE = SUM/25.0
WRITE(6, 20) AVE
20 FORMAT(1X, 'THE CLASS AVERAGE IS', 1X, F4.1)
  
```

يمكن حساب متوسط الفصل الدراسي بطريقة مرتبة باستخدام خريطة سير العمليات وجزء البرنامج في شكل ٦ - ٢ . (قارن مع شكل ٤ - ١٧) . ومع ذلك لا توجد طريقة لمقارنة أى من الدرجات بمتوسط الفصل بمأن الدرجات لم تخزن في الذاكرة . وبذلك لا يمكن إيجاد عدد الطلبة الذين حصلوا على درجات أقل من المتوسط .

استخدام المتغيرات ذات الأدلة يعتبر ضرورياً للغاية إذا أردنا استخدام المخطط السابق ، ويكون في استطاعتنا حساب عدد الطلبة الذين أخذوا درجات أقل من متوسط الفصل . بالتحديد ، تخزن الدرجات في مجموعة متراسة خطية إسمها SCORE بها 25 عنصر . تظهر خريطة سير العمليات ومايقابلها بالفورتران في شكل ٦ - ٣ .

ملاحظة : نخدم أول جملة WRITE في شكل ٦-٣ غرضين ، فإلى جانب أنها تطلع الدرجات فهي تستخدم أيضاً كاختبار لاكتشاف ما إذا كانت المجموعة المتراسة SCORE قد قرئت بطريقة صحيحة أم لا . يسمى هذا الاختبار باختبار الصدى - ونلاحظ أن اختبار الصدى وريته مفيدة للغاية في أحيان كثيرة وتستخدم في اكتشاف الأخطاء ، أى تحديد أخطاء البرنامج . تضاف في أحيان كثيرة جملة WRITE للبرامج المعقدة لتظهر نتائج وسيطة ، عندئذ ، يمكن تحديد وإيجاد الأخطاء بسهولة . وبعد اكتشاف كل الأخطاء (bugs أو errors) ، يمكن حينئذ إلغاء هذه الجملة . (رغم أن أول جملة READ و WRITE تظهر في حلقة DO التكرارية الخاصة بهم ، فن الراضع أنه كان من الممكن إدماجهم مع حلقة DO الثانية وجعلها حلقة واحدة) .

ندرس الآن المسألة الأصلية للفصل الدراسي الذي لا يزيد عدد طلبته عن 25 طالباً . ولما كان المطلوب معرفة عدد الطلبة N الذين أدوا الامتحان لإيجاد متوسط الفصل ، فإنه يجب أن تعرف N في البرنامج . نعتبر حالتين في إحدى هاتين الحالتين تشمل مجموعة البيانات بطاقة مقدمة ، وفي الأخرى يضاف لمجموعة البيانات بطاقة خلفية (تذكر قسم ٤ - ١١ من أجل مناقشة بطاقات المقدمة والبطاقات الخلفية) .

حالة (أ)

افترض أن مجموعة البيانات لها بطاقة مقدمة تحتوي على عدد الطلبة (N) المؤدين للامتحان . يمكن تعديل البرنامج في شكل ٦ - ٣ بسهولة كما يلي : أضف الآتي بعد جملة DIMENSION

```

READ(5, 40) N
40  FORMAT(I3)
WRITE(6, 50) N
50  FORMATE('1', 'NUMBER TAKING EXAM IS', 1X, I3)

```

ثم بعد ذلك غير 25 بـ N في جملة DO الثلاث ، وغير $AVE = SUM/25.0$ إلى :

```
AVE = SUM/FLOAT(N)
```

فيما عدا ذلك ، فالبرنامج مطابق للبرنامج السابق .

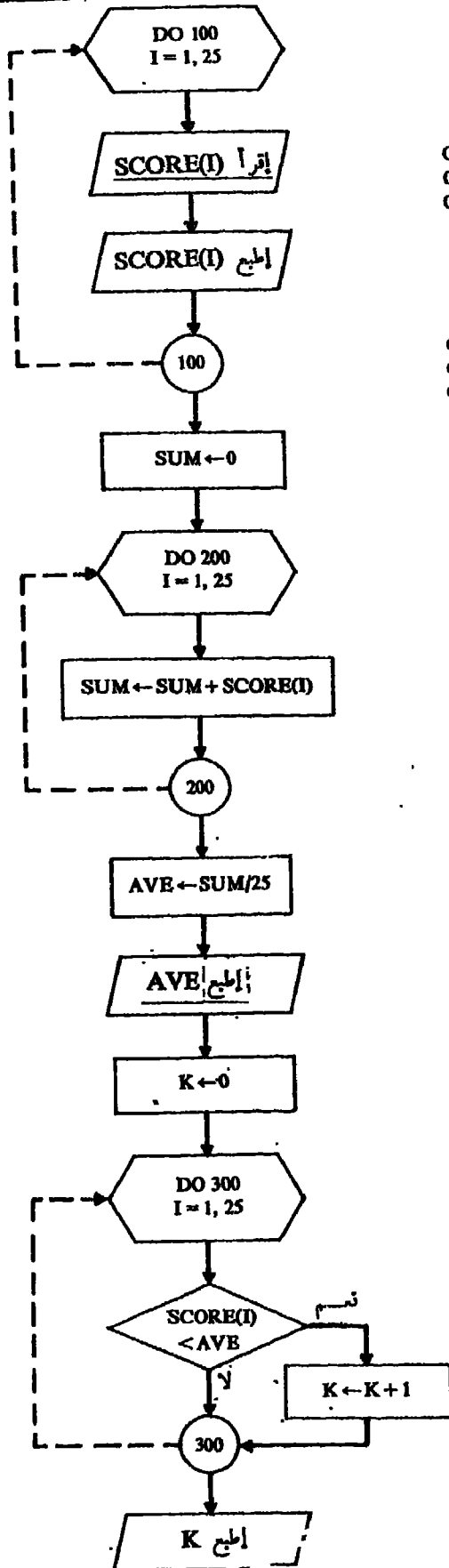
حالة (ب)

افترض أن مجموعة البيانات لها بطاقة خلفية تحتوي على رقم سالب . يخزن جزء برنامج الفورتران التالي الدرجات في المجموعة المتراسة SCORE ويحسب أيضاً عدد الطلبة الذين أدوا الامتحان (N) :

```

I = 0
111 READ(5, 60) A
60  FORMAT(F10.1)
IF(A.LT.0.0) GO TO 222
I = I + 1
SCORE(I) = A
GO TO 111
222 N = I

```



```

DIMENSION SCORE(25)
C
C
C
READ IN SCORES
DO 100 I = 1, 25
    READ(5, 10) SCORE(I)
    WRITE(6, 10) SCORE(I)
    FORMAT(F10.1)
10
100 CONTINUE
C
C
C
COMPUTE THE SUM AND AVERAGE
SUM = 0.0
DO 200 I = 1, 25
    SUM = SUM + SCORE(I)
200 CONTINUE
AVE = SUM/25.0
WRITE(6, 20) AVE
20 FORMAT(1X, 'CLASS AVERAGE IS', 1X, F10.1)
C
C
C
COUNT THE NUMBER BELOW AVERAGE
K = 0
DO 300 I = 1, 25
    IF(SCORE(I).LT.AVE) K = K + 1
300 CONTINUE
WRITE(6, 30) K
30 FORMAT(1X, 'NUMBER BELOW AVERAGE IS', 1X, I3)
STOP
END
    
```

شكل ٦ - ٣

يمكن عمل هذا أيضاً باستخدام حلقة DO . نجعل الدليل يتراوح ما بين 1 إلى 26 حيث لا يمكن أن يكون هناك أكثر من 25 درجة مع البطاقة الخلفية

```
DO 500 I = 1, 26
  READ(5, 60) A
60  FORMAT(F10.1)
  IF(A.LT.0.0) GO TO 222
  SCORE(I) = A
500 CONTINUE
222 N = I - 1
```

لاحظ أن $N = I - 1$ حيث أن الدليل I يجب أيضاً البطاقة الخلفية . حيث أن هناك 25 طالباً على الأكثر ، فيكون هناك دائماً مخرج غير طبيعي من حلقة DO . وعلى ذلك تتحدد دائماً قيمة I في الجملة 222 .
فيما يلي أمثلة أخرى تشتمل على مجموعات متراسة .

مثال ٦ - ١

افرض أن $A(1), A(2), \dots, A(N)$ مخزنة في الذاكرة وأن $N \geq 2$. والمطلوب إيجاد أكبر الأعداد للمجموعة المتراسة A .
(أ) إحدى هذه الطرق هي إيجاد القيمة الحالية للرقم الأكبر أثناء المرور خلال المجموعة المتراسة (كما سبق أن ناقشنا ذلك في م - ١٠) .

```
XLAR = A(1)
DO 100 I = 2, N
  IF(A(I).LE.XLAR) GO TO 100
  XLAR = A(I)
100 CONTINUE
```

(ب) الطريقة الثانية مفيدة عند فرز البيانات وفيها يتم تحديد مكان L للقيمة العظمى وبذا تكون $A(L)$ هي القيمة العظمى . هذا النظام الحسابي (الخوارزم) مشابه للنظام السابق فيما عدا أننا الآن نتذكر المكان L للعدد الأكبر الحال بدلاً من العدد نفسه .

```
L = 1
DO 200 I = 2, N
  IF(A(I).LE.A(L)) GO TO 200
  L = I
200 CONTINUE
XLAR = A(L)
```

في الحالة التي لانعرف فيها أن $N \geq 2$ (أي ، N يمكن أن تساوي 1) ، فإنه يجب تغيير جملة DO بحيث يتراوح الدليل I من 1 إلى N .

(هل يستطيع القارئ شرح ما هو الفرق ؟)

مثال ٦ - ٢

اعتبر أن $A(1), A(2), \dots, A(N)$ مخزنة في الذاكرة . استعمل حلقة DO لطباعة $A(N), A(N-1), \dots, A(2), A(1)$ بحيث تكون كل واحدة في سطر (أنظر مثال ٥ - ٥ (ج))

كما سبق وأن ناقشنا من قبل فلا يمكن أن يكون معامل الزيادة حلقة DO غير الهيكلية سالبة ، ومن ثم فالبرنامج التالي غير صحيح :

```
DO 300 I = N, 1, -1
    WRITE(6, 10) A(I)
10    FORMAT(10X, F10.1)
300 CONTINUE
```

وعلى أي حال لما كان الدليل I يتغير من 1 إلى N ، ونريد أن نطبع A(N) ، ... ، A(1) أي لدينا :

قيمة الدليل I	العنصر الذي سيُطبع
1	A(N)
2	A(N - 1)
3	A(N - 2)
...	...
N	A(1)

هناك علاقة محددة بين الدليل I ودليل A فمجموعها دائماً $N + 1$ وبذلك سيُطبع البرنامج التالي عناصر A في الترتيب العكسي كما هو مطلوب :

```
DO 300 I = 1, N
    K = N + 1 - I
    WRITE(6, 10) A(K)
10    FORMAT(10X, F10.1)
300 CONTINUE
```

٦ - ٦ المجموعات المتراسة ذات الأبعاد المتعددة

المجموعات المتراسة الخطية (ذات البعد الواحد) والتي تمت مناقشتها حتى الآن لها دليل واحد فقط ، ويسمح الفورتران أيضاً بمجموعات متراسة لها دليان أو ثلاثة أدلة . تسمى هذه مجموعات متراسة ذات بعدين أو ثلاثة أبعاد ، على الترتيب . في الحقيقة تسمح بعض الحاسبات الكبيرة بمجموعات متراسة لها سبعة أدلة . وفي هذا القسم ، سنتناقص هذه المجموعات المتراسة ذات الأبعاد المتعددة .

تعرف المصفوفة رياضياً كمجموعة متراسة على شكل مستطيل من الأعداد فثلاثاً ما يلي :

أول صف	a_{11}	a_{12}	a_{13}	a_{14}
ثاني صف	a_{21}	a_{22}	a_{23}	a_{24}
ثالث صف	a_{31}	a_{32}	a_{33}	a_{34}
	أول عمود	ثاني عمود	ثالث عمود	رابع عمود

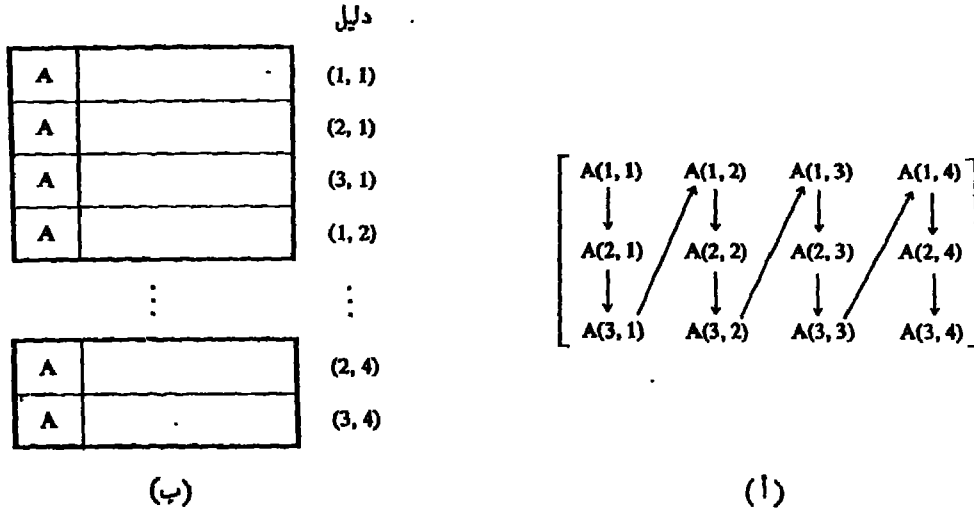
مصفوفة 3 في 4 (وتكتب 3×4) أي ، مصفوفة بها ثلاثة صفوف (الأسطر الأفقية للأعداد) وأربعة أعمدة (الأسطر الرأسية للأعداد) تعرف على كل عنصر في المصفوفة بعددين صحيحين ، هما أدلة . يعرف الدليل الأول مكان العنصر في الصف والدليل الثاني مكانه في العمود . على سبيل المثال ، تظهر a_{22} في الصف الثاني والعمود الأول ، وتظهر a_{23} في الصف الثاني والعمود الثالث ، وهكذا . وتكتب عادة 3×4 بدلا من 3 في 4 ("3 by 4")

وبطريقة مناظرة ، في الفورتران المجموعة المتراسة ذات بعدين $(m \times n)$ هي قائمة بها عدد $m.n$ من خلايا الذاكرة (عناصر المجموعة المتراسة) حيث يمكن أن يتحدد أي عنصر معين بزواج من الأعداد الصحيحة K و L حيث $1 \leq K \leq m$ و $1 \leq L \leq n$. والأعداد الصحيحة K و L تسمى أدلة العنصر . الدليل الأول K يسمى صف العنصر والدليل الثاني L عمود العنصر . (مثل هذه المجموعات المتراسة ذات - بعدين ستسمى أيضاً مجموعات متراسة أو مصفوفة) .

على سبيل المثال ، افرض A هي اسم مجموعة متراسة ذات بعدين (3×4) . إذن A يكون بها $3 \times 4 = 12$ عنصراً .

$$A(1, 1), A(1, 2), A(1, 3), A(1, 4), A(2, 1), A(2, 2), A(2, 3), \dots, A(3, 3), A(3, 4)$$

لاحظ أن كل عنصر يحدد باسم المجموعة المتراسة يتيمه زوج من الأدلة . مفصولان بفصلة ومحاطان بأقواس . مع أن A تصور عادة كجموعة متراسة في شكل مستطيل كما في الشكل ٦ - ٤ (أ) ، إلا أنها تستعمل داخلياً في الحاسب على أنها سلسلة من 12 خلية ذاكرة كما في الشكل ٦ - ٤ (ب) . لاحظ أن العناصر تخزن عمودية ، حتى أن الدليل الأول (صف) يتغير أولاً ثم يتغير الدليل الثاني (عمود) . وترتيب هذا التخزين موضح بالأسهم في شكل ٦ - ٤ (أ)



شكل ٦ - ٤

بالمثل ، المجموعة المتراسة ذات الأبعاد الثلاثة $(m \times r \times s)$ هي قائمة بها عدد $m.r.s$ من خلايا الذاكرة (عناصر المجموعة المتراسة) وفيه يتحدد عنصر معين بثلاثة أعداد صحيحة J, K, L (الأدلة) ، حيث

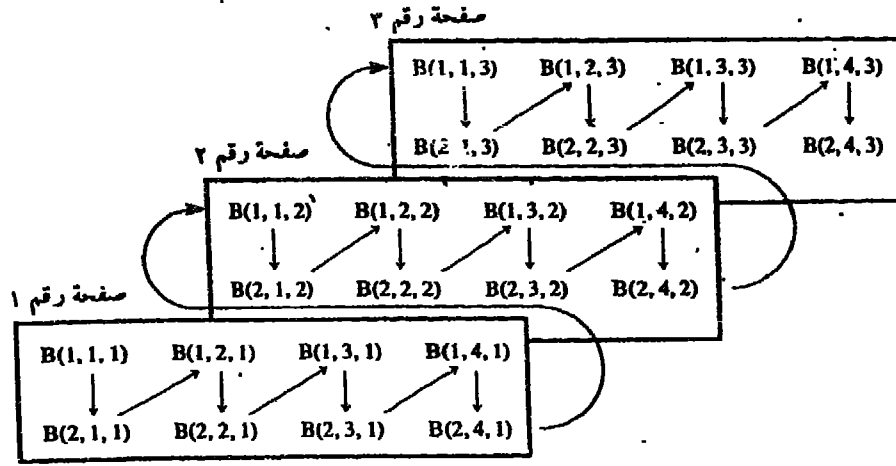
$$1 \leq J \leq m, \quad 1 \leq K \leq r, \quad 1 \leq L \leq s$$

على سبيل المثال ، افرض أن B مجموعة متراسة ذات ثلاثة أبعاد $(2 \times 4 \times 3)$ إذن B يكون بها $3 \times 4 \times 2 = 24$ عنصراً كما تظهر في شكل ٦ - ٥ . لاحظ أن B قد صورت كأنها مكونة من ثلاث طبقات (صفحات) من مصفوفة 2×4 يسمى الدليل الأول الصف والدليل الثاني العمود ، والدليل الثالث صفحة العنصر ، على سبيل المثال $B(1, 4, 2)$ هي العنصر في الصف الأول ، العمود الرابع والصفحة الثانية من B . ومع ذلك ، تمثل B داخلياً كسلسلة من 24 خلية ذاكرة مرصوفة بحيث يتغير الدليل الأول أولاً (أكثر سرعة) ، والدليل الثاني ثانياً ، وهكذا ، كما في الشكل ٦ - ٦ . أي أن العناصر تنظم عمودياً من أول صفحة ثم عمودياً من ثاني صفحة ، ثم عمودياً من ثالث صفحة ويوضح ترتيب التخزين بالأسهم كما في شكل ٦ - ٥ .

وعلى العموم فالهجوم المتراسة ذات عدد p من الأبعاد ، أي $(n_1 \times n_2 \times \dots \times n_p)$ هي قائمة بها $n_1.n_2.\dots.n_p$ عنصراً من خلايا الذاكرة (عناصر المجموعة المتراسة) وفيه يتحدد عنصر معين بواسطة p من الأعداد الصحيحة K_1, K_2, \dots, K_p (الأدلة) حيث

$$1 \leq K_1 \leq n_1, \quad 1 \leq K_2 \leq n_2, \quad \dots, \quad 1 \leq K_p \leq n_p$$

يحدد عنصر معين باسم المجموعة المتراسة متبوعاً بأدلة p مفصولة عن بعضها بواسطة فاصلات ومحاطة بأقواس .



شكل ٦ - ٥

الأدلة	
B	(1, 1, 1)
B	(2, 1, 1)
B	(1, 2, 1)
B	(2, 2, 1)
B	(1, 3, 1)
⋮	⋮
B	(2, 3, 3)
B	(1, 4, 3)
B	(2, 4, 3)

شكل ٦ - ٦

المجموعات المتراسة ذات الأبعاد المتعددة يجب أن تعرف بنفس طريقة المجموعات المتراسة ذات بعد واحد على سبيل المثال :
DIMENSION ID(25), MONEY(3, 2, 7), TAX(6, 5)

ستبلغ المترجم أن :

١ - ID مجموعة متراسة ذات بعد واحد ويحتمل أن يكون له 25 عنصراً (أى ، سيخصص له 25 خلية ذاكرة مع عدم الالتزام باستخدام كل الخلايا بالبرنامج) .

٢ - MONEY مجموعة متراسة ذات ثلاثة أبعاد ويحتمل أن يكون بها $3 \times 2 \times 7 = 42$ عنصراً .

٣ - TAX مجموعة متراسة ذات بعدين ويحتمل أن يكون لها $6 \times 5 = 30$ عنصراً .

إننا نؤكد أن أسماء المجموعات المتراسة متعددة الأبعاد والأشكال لتناسرها تتبع نفس قوانين المجموعات المتراسة ذات البعد الواحد .

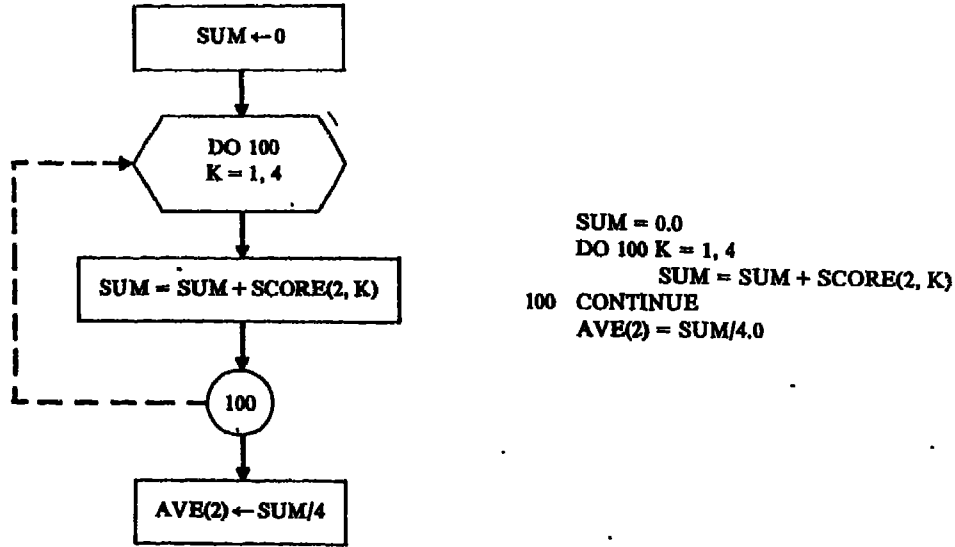
مثال ٦ - ٣

يفرض أنك أعطيت 4 اختبارات لفصل به 25 طالباً . يمكن أن تنظم درجات الاختبارات هذه في جدول كما يلي :

طالب	اختبار رقم ١	اختبار رقم ٢	اختبار رقم ٣	اختبار رقم ٤
1	78	83	80	87
2	100	100	92	100
⋮	⋮	⋮	⋮	⋮

إفرض أن درجات الاختبار هذه تم تخزينها في مجموعة متراصة حقيقية في شكل مصفوفة SCORE أبعادها (25×4) . وبذلك SCORE (I, K) تعطي درجة الطالب I في الاختبار K . بالتحديد ، فالصف الثاني :

يعطي درجات الاختبارات الأربعة للطالب الثاني . لترجمة متوسط الطالب ، سنجعل AVE(I) تشير إلى متوسط اختبار الطالب I يمكن أن نستخدم خريطة سير العمليات في شكل ٦ - ٧ ومايقابلها بالفورتران لإيجاد AVE(2) ، أي متوسط درجات الطالب الثاني .



شكل ٦ - ٧

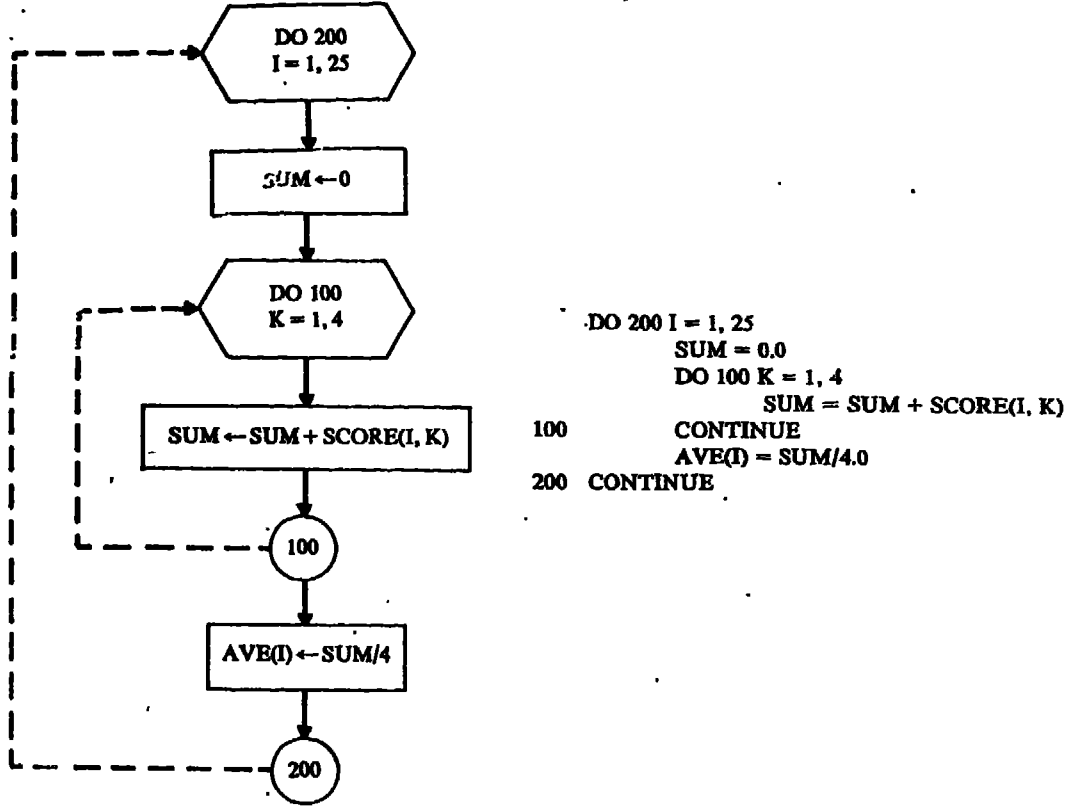
وبصورة عامة يمكن أن نجد AVE(I) متوسط درجات الطالب I بجزء من برنامج الفورتران التالي :

```

SUM = 0.0
DO 100 K = 1, 4
SUM = SUM + SCORE(I, K)
100 CONTINUE
AVE(I) = SUM/4.0

```

ومن ثم لحساب متوسط كل طالب من الـ 25 طالباً . سنحتاج إلى حلقة DO متداخلة حيث حلقة DO الخارجية تنفذ 25 طالباً . خريطة سير العمليات تظهر في الشكل ٦ - ٨ وكذلك مايقابلها بالفورتران .



شكل ٦ - ٨

٦ - ٧ ادخال / اخراج مجموعة متراسة وحلقات DO الضمنية

افرض A و B مجموعات متراسة خطية مكونة من أربعة وثلاثة عناصر ، على الترتيب . وكما ذكرنا سابقا ، يمكن أن تظهر عناصر A و B في جملة I/O (إدخال / إخراج) كتغيرات عادية . على سبيل المثال :

```
READ(5, 10) A(3), B(2), J, X
```

تخبر الحاسب أن يقرأ قيم لـ A(3) و B(2) وكذلك J و X . بالتحديد يمكن أن ندخل المجموعات المتراسة A و B بأكملها بذكر كل عناصرها مفردة ، على سبيل المثال :

```
READ(5, 20) A(1), A(2), A(3), A(4), B(1), B(2), B(3)
```

واضح أن هذا الأسلوب الفني لن يكون مناسباً إذا كان هناك مئات من العناصر في المجموعات المتراسة . سنناقش الآن طرقاً أخرى لإدخال / إخراج المجموعات المتراسة .

(١) أسماء المجموعات المتراسة في جملة I/O (الإدخال / الإخراج) :

يمكن أن نقرأ أو نطبع مجموعة متراسة بأكملها باستخدام اسم المجموعة المتراسة وذلك في جملة الإدخال / الإخراج فقط . على سبيل المثال ، يمكن أن نعيد كتابة جملة READ السابقة بطريقة مكافئة .

```
READ(5, 20) A, B
```

ومع ذلك ، يجب أن نفهم بوضوح ما يترتب على أمر I/O مثل هذا :

١ - أولاً سيخزن أو سيطبع المجموعة المتراسة بأكملها أى أن كل أماكن الذاكرة المخصصة بجملته DIMENSION ستأخذ قيمة . وتبعاً لذلك ، إذا عرفت SCORE كمجموعة متراسة كما يلي :

DIMENSION SCORE(35)

إذن يجب ألا نستخدم :

READ(5, 30) SCORE
30 FORMAT(SF10.1)

إذا أردنا أن نقرأ 25 قيمة فقط داخل SCORE .

٢ - ستخصص أو ستطبع قيم لعناصر المجموعة المتراسة تبعاً لترتيب تخزين المجموعة المتراسة (شكل ٦-١ ، ٦-٤ ، ٦-٦) ، حيث يتغير الدليل الأول بسرعة أكبر ، والدليل الثانى بعده ، وهكذا .

مثال ٦-٤

إنرض E و D و C مجموعات متراسة معرفة بالجملته :

DIMENSION C(3), D(2, 3, 2), E(3, 2)

إذن فزوج الجمل WRITE-FORMAT

WRITE(6, 30) C, D, E
30 FORMAT(6X, 4F10.2)

تأمر الحاسب أن يطبع مجموع القيم التالية :

$$3 + 2 \cdot 3 \cdot 2 + 3 \cdot 2 = 21$$

أى 3 من المجموعة المتراسة C و 12 من المجموعة المتراسة D و 6 من المجموعة المتراسة E . حيث أن جملة FORMAT لها أربعة مواصفات حقول رقمية فقط ، ستطبع أربعة أرقام على كل سطر ، وستستعمل جملة FORMAT مرة أخرى وأخرى إلى أن تم طباعة كل القيم . وبذلك سيبدو الخرج كما فى الشكل ٦-٩ .

C(1)	C(2)	C(3)	D(1, 1, 1)
D(2, 1, 1)	D(1, 2, 1)	D(2, 2, 1)	D(1, 3, 1)
D(2, 3, 1)	D(1, 1, 2)	D(2, 1, 2)	D(1, 2, 2)
D(2, 2, 2)	D(1, 3, 2)	D(2, 3, 2)	E(1, 1)
E(2, 1)	E(3, 1)	E(1, 2)	E(2, 2)
E(3, 2)			

شكل ٦-٩

تذكر أنه إذا استعملت جملة FORMAT مختلفة فى المثال السابق وليكن :

30 FORMAT(6X, 6F10.2)

فسوف يشمل الخرج ستة أرقام على كل سطر . ومع ذلك لن يتغير ترتيب الأرقام بجملته FORMAT المختلفة .

(ب) حلقات DO الضمنية

إفرض أننا نريد أن نخزن أو نطبع جزءاً من مجموعة متراسة أو افرض أننا لانريد أن نخزن أو نطبع مجموعة متراسة بتقريبها الموصوف في التخزين . قد يمكننا عمل هذا باستخدام حلقة DO . على سبيل المثال ، إفرض أن AMOUNT مجموعة متراسة خطية محتمل أن تحتوي على 100 عنصر . إذن فحلقة DO التالية :

```
DO 100 I = 11, N, 1
  READ(5, 10) AMOUNT(I)
100 CONTINUE
```

تأمر الحاسب أن يقرأ قسِم (AMOUNT(11) ، AMOUNT(12) ، حتى ومتفصلاً AMOUNT(N) ، حيث N تم تعريفها مسبقاً و $N \leq 100$. يمكن أن نتجز نفس الشيء باستخدام جملة READ التالية :

```
READ(5, 20) (AMOUNT(I), I = 11, N, 1)
```

التعبير

```
(AMOUNT(I), I = 11, N, 1)
```

يسمى حلقة DO الضمنية حيث أنها تسبب حدوث حلقة تكرارية بداخل جملة READ والرمز :

```
I = 11, N, 1
```

له نفس المعنى هنا كما هو في حلقة DO ، أى ، 11 هي القيمة الابتدائية و N هي القيمة النهائية ، و 1 هو مقدار الزيادة .

تذكر أيضاً أن حلقة DO الضمنية أكثر تعبيراً من حلقة DO السابقة . بالتحديد كل مرة تصادف جملة READ في حلقة DO السابقة يجب أن تقرأ بطاقة بيانات أخرى بغض النظر عن كيفية تحديد جملة FORMAT المرقمة 10 . وتباً لذلك يجب وضع قيم المدخلات كل على بطاقة منفصلة . من ناحية أخرى ، فإننا نمر ، في حلقة DO الضمنية ، على جملة READ واحدة فقط ، وبذلك يمكن وضع المدخلات على بطاقة أو أكثر ممتداً في ذلك على جملة FORMAT .

فيما يلي الشكل العام لحلقة DO الضمنية لجملة I/O .

```
(VN(I), I = IN, IE, IC)
```

حيث :

IE قيمة الاختيار أو النهاية لـ I

VN اسم المجموعة المتراسة

IC معامل الزيادة

I الدليل

IN القيمة الابتدائية لـ I

تذكر أن كلا من IN ، IE و IC يمكن أن يكون عدداً صحيحاً موجباً أو متبديلاً صحيحاً . أيضاً ، إذا كانت قيمة IC هي 1 فإنه يمكن حذفها . وهنا نؤكد أن الأقواس الخارجية والفصلة قبل الدليل ضرورية للغاية عند استخدام حلقة DO الضمنية .

يمكن استخدام الدليل I للسيطرة على أكثر من مجموعة متراسة واحدة . على سبيل المثال :

```
READ(5, 30) (A(K), B(3, K), C(K, K, 2), K = 2, 9, 3)
```

تغير الحاسب أن يدور خلال قائمة المتغيرات أولاً عندما $K = 2$. ثم عندما $K = 5$ ، وأخيراً عندما $K = 8$ ، مز ثم يكون جملة READ مكافئة لـ :

```
READ(5, 30) A(2), B(3, 2), C(2, 2, 2), A(5), B(3, 5), C(5, 5, 2),
1          A(8), B(3, 8), C(8, 8, 2)
```

عموماً ليس من الضروري أن تكون المتغيرات في حلقة DO الضمنية ذات دليل أو تتضمن العداد .

مثال ٦ - ٥

(أ) افرض X مجموعة متراسة خطية بها 100 عنصر مخزنة فعلاً في الذاكرة يمكن أن تطبع المجموعة المتراسة كل عنصر منها على سطر بواسطة :

```
WRITE(5, 20) (X(I), I = 1, 100)   - DO 100 I = 1, 100
20 FORMAT(2X, F8.2)              WRITE(5, 10) X(I)
                                  10 FORMAT(2X, F8.2)
                                  100 CONTINUE
```

يمكن أن تطبع المجموعة المتراسة كل أربعة عناصر منها على سطر بواسطة :

```
DO 100 I = 1, 100, 4
    WRITE(5, 10) X(I), X(I + 1), X(I + 2), X(I + 3)
10    FORMAT(4(2X, F8.2))
100 CONTINUE
```

```
WRITE(5, 20) (X(I), I = 1, 100)
20 FORMAT(4(2X, F8.2))
```

وبذلك نرى سهولة تغيير حلقة DO الضمنية عن حلقة DO لعمليات I/O .

(ب) اعتبر زوج الجمل WRITE-FORMAT التالي :

```
WRITE(6, 40) (A(I), B, X(3), Y(I, I), I = 1, 8, 3)
40 FORMAT(6(2X, F8.2))
```

تنفذ حلقة DO داخل جملة WRITE ثلاث مرات ، أولاً مع $I = 1$ ثم $I = 4$ وأخيراً مع $I = 7$ وعلى ذلك تكون جملة WRITE مكافئة لما يلي :

```
WRITE(6, 40) A(1), B, X(3), Y(1, 1), A(4), B, X(3),
1          Y(4, 4), A(7), B, X(3), Y(7, 7)
```

لاحظ أن B و $X(3)$ يظهران ثلاث مرات . ستسبب جملة FORMAT طبع ستة أرقام في السطر ، بحيث يبدو المخرج كالتالي :

```
A(1)  B      X(3)  Y(1, 1)  A(4)  B
X(3)  Y(4, 4) A(7)  B      X(3)  Y(7, 7)
```

(ج) سيطبع جزء البرنامج التالي :

```
WRITE(6, 50) (I, X(I), I = 1, 3)
50 FORMAT(1X, 'A(', I1, ') = .', F10.2)
```

ما يلي :

```
A(1) = XXXXXXXX.XX
A(2) = XXXXXXXX.XX
A(3) = XXXXXXXX.XX
```

(ح) حلقات DO الضمنية المتداخلة

حلقة DO الضمنية الموصوفة أعلاه يمكن أن تميم لتسيطر على مجموعات متراسة متعددة الأبعاد (هذا التعميم مشابه لحلقات DO المتداخلة) . إفرض ، على سبيل المثال ، أن F قد عرفت على أنها مجموعة متراسة (3×2) والبرنامج يحتوى على الجملة

READ(5, 10) ((F(I, J), J = 1, 2), I = 1, 3)

فالدليل الخارجى I مشابه لدليل حلقة DO الخارجية ، والدليل الداخلى J مشابه لدليل حلقة DO الداخلية . أى ، تنفيذ حلقة DO الضمنية الداخلية بالكامل لكل قيمة من دليل حلقة DO الضمنية الخارجية . وبذلك تكون جملة READ مكافئة ل :

READ(5, 10) (F(1, J), J = 1, 2), (F(2, J), J = 1, 2), (F(3, J), J = 1, 2)

التي هي مكافئة ل :

READ(5, 10) F(1, 1), F(1, 2), F(2, 1), F(2, 2), F(3, 1), F(3, 2)

لاحظ أن جملة READ العليا تقرأ القسم إلى داخل F صفاً بصف . من ناحية أخرى الجملة :

READ(5, 20) ((F(I, J), I = 1, 3), J = 1, 2)

(لاحظ أنه تم تبديل الأداة I ، J) تكافئ الجملة

READ(5, 20) F(1, 1), F(2, 1), F(3, 1), F(1, 2), F(2, 2), F(3, 2)

وفى هذه الحالة ، تقرأ القسم إلى داخل F عموداً بعمود .

يمكن أن يكون لحلقات DO المتداخلة تركيبات أكثر تعقيداً ويمكن أن تظهر أيضاً فى جمل WRITE بنفس الطريقة .

مثال ٦ - ٦

(أ) إدرس الجملة

WRITE(6, 11) ((B(L, M), L = 2, 10, 2), M = 1, 11, 3)

أولاً ، تنفيذ حلقة DO الداخلية $(B(L, M), L = 2, 10, 2)$ عندما يكون $M = 1$ ، $M = 4$ ، $M = 7$ ، وأخيراً $M = 10$ هكذا تكون الجملة مكافئة ل :

WRITE(6, 11) (B(L, 1), L = 2, 10, 2), (B(L, 4), L = 2, 10, 2),
1 (B(L, 7), L = 2, 10, 2), (B(L, 10), L = 2, 10, 2)

وهي مكافئة ل :

WRITE(6, 11) B(2, 1), B(4, 1), B(6, 1), B(8, 1), B(10, 1), B(2, 4), B(4, 4),
1 B(6, 4), B(8, 4), B(10, 4), B(2, 7), B(4, 7), B(6, 7), B(8, 7),
1 B(10, 7), B(2, 10), B(4, 10), B(6, 10), B(8, 10), B(10, 10)

لاحظ أن الدليل الثانى M فى $B(L, M)$ يظهر فى حلقة DO الخارجية ، بينما يظهر الدليل الأول I فى حلقة DO الخارجية فى المثال السابق .

(ب) إفرض B مجموعة متراسة $(3 \times 5 \times 4)$ وأفرض أننا نريد أن نطبع قيم B من الصفحة الأولى صففاً بصف ثم من الصفحة الثانية صففاً بصف وأخيراً من الصفحة الأخيرة صففاً بصف . يمكن أن يتم هذا بزواج الجمل WRITE-FORMAT :

WRITE(6, 50) ((B(K1, K2, K3), K2 = 1, 5), K1 = 1, 4), K3 = 1, 3)
50 FORMAT(5(2X, F8.2))

فى الحقيقة ، ستسمح جملة FORMAT بطباعة خمسة أرقام فقط على السطر ، بحيث تظهر الصفوف المختلفة ل B على أسطر مختلفة من صفحة الطباعة .

٦ - ٨ مثال لأسلوب البرمجة الفني

يمكن أن يكون هناك عدة أنظمة حسابية (خوارزميات) لحل وصياغة أى مسألة ويكون البعض أفضل من البعض الآخر ، أى ، يحتاج البعض عدد خذيا ذاكرة أقل و / أو عمليات أقل . ونكسر الفصل الثامن لعدة أساليب للبرمجة الفنية . وندرس هنا مسألة بسيطة توضح النقط الدقيقة الماكورة التي يمكن أن تبرز أثناء الحل .

إفرض أننا نريد أن نحرك العناصر $A(1)$ ، $A(2)$ ، ... ، $A(N)$ في مجموعة متزايدة ما أسفله القائمة مكاناً واحداً :

$$A(1) \rightarrow A(2) \rightarrow A(3) \rightarrow \dots \rightarrow A(N-1) \rightarrow A(N)$$

أى ، حتى تحتوي $A(2)$ قيمة $A(1)$ القديمة ، وتحتوي $A(3)$ قيمة $A(2)$ القديمة وهكذا . فالشيء اليدهى أن نبدأ بتخصيص $A(1)$ إلى $A(2)$ ثم $A(2)$ إلى $A(3)$ وهكذا إلى أن نخصص $A(N)$ إلى $A(N+1)$. يمكن عمل هذا بحلقة DO التكرارية :

```
DO 100 I = 1, N
    A(I + 1) = A(I)
100 CONTINUE
```

ومع ذلك لن يتم تنفيذ هذا الإجراء . السبب هو أنه عند تخصيص $A(1)$ إلى $A(2)$ في بادئ الأمر تم إزالة القيمة الأصلية لـ $A(2)$ وبذلك تم تخصيص $A(2)$ التالي إلى $A(3)$ سيخزن القيمة الأصلية لـ $A(1)$ في $A(3)$ - ومن ثم فبواسطة الإجراء السابق فإن القيمة الأصلية لـ $A(1)$ ستخصص إلى كل أماكن التخزين $A(2)$ ، $A(3)$ ، ... ، $A(N+1)$.

والطريقة لتصحيح الإجراء السابق هي أن نخزن القيم الأصلية لـ A مؤقتاً في مراكز تخزين أخرى . يمكن عمل هذا باستخدام مجموعة متزايدة أخرى B . بالتحديد ، نستخدم حلقة DO لتخصيص $A(I)$ إلى $B(I+1)$ بعد ذلك نستخدم حلقة DO أخرى لنقل العناصر إلى A مرة أخرى بتخصيص $B(I+1)$ إلى $A(I+1)$. وذلك يعنى أن نستخدم ضعف عدد خلايا الذاكرة التي في A لمجرد نقل عناصر A إلى أسفل مكان واحد ، أى إذا احتلت A 1000 خلية ذاكرة إذن سنحتاج 1000 خلية ذاكرة أخرى لـ B .

طريقة أخرى ، نحتاج فيها إلى خليتين إضافيتين من خلايا الذاكرة فقط ، نخزن أولاً $A(2)$ في مكان مؤقت $TEMPA$ قبل تخصيص $A(1)$ إلى $A(2)$ ثم نخزن $A(3)$ في مكان مؤقت $TEMPB$ قبل تخصيص $TEMPA$ إلى $A(3)$

```
TEMPA ← A(2)
A(2) ← A(1)
TEMPB ← A(3)
A(3) ← TEMPA
```

لاحظ أن $A(3)$ تحتوي الآن $A(2)$ الأصلية . ثم نستعمل بعد ذلك $TEMPA$ مرة أخرى لتخزين $A(4)$ قبل تخصيص $TEMPB$ إلى $A(4)$ وهكذا . يمكن إنجاز مسبق بحلقة DO التكرارية . رغم أننا استخدمنا خليتين إضافيتين للذاكرة فقط إلا أن النظام الحسابي (الخوارزم) يحتاج فعلاً إلى $2N - 1$ تخصيصاً .

في الحقيقة أن أفضل طريقة لا تحتاج أى خلية ذاكرة إضافية وتحتاج عدد N تخصيصات فقط . وهي أننا ، نحرك ببساطة $A(N)$ إلى $A(N+1)$ ثم $A(N-1)$ إلى $A(N)$. وهكذا أى نعطي سلسلة الأوامر التالية :

```
A(N + 1) ← A(N)
A(N) ← A(N - 1)
⋮
A(2) ← A(1)
```

فما يلي الفورتران المكافئ :

```
DO 100 J = 1, N
    A(N - J + 2) = A(N - J + 1)  أوببساطة
100 CONTINUE
DO 100 J = 1, N
    K = N + 1 - J
    A(K + 1) = A(K)
100 CONTINUE
```

(أنظر المثال ٦ - ٢ ترى السبب في استخدام المتغير الجديد K)

مسائل محلولة

مجموعات متراسة

٦ - ١ حدد عدد العناصر في المجموعات المتراسة تباعاً لجملة DIMENSION التالية :

DIMENSION A(25), B(3, 8, 4), K(7, 5)

A بها 25 عنصراً .

B بها $3 \times 8 \times 4 = 96$ عنصراً .

C بها $7 \times 5 = 35$ عنصراً .

٦ - ٢ أوجد عدد العناصر في المجموعة المتراسة X المعرفة بالآتي :

DIMENSION X(2, 4, 3)

صف الخرج إذا نفذنا الجملتين الآتيتين :

WRITE(6, 10) X
10 FORMAT(6(2X, F12.2))

X بها $2 \times 4 \times 3 = 24$ عنصراً . ستطبع كما هي مخزنة ، اي ، بحيث يتغير الدليل الأول (صف) بأقصى سرعة ، يليه الدليل الثاني (عمود) وأخيراً الدليل الثالث (صفحة) . أيضاً ، سيكون هناك ستة أرقام في السطر . وبذلك يظهر الخرج كالتالي :

X(1, 1, 1)	X(2, 1, 1)	X(1, 2, 1)	X(2, 2, 1)	X(1, 3, 1)	X(2, 3, 1)
X(1, 4, 1)	X(2, 4, 1)	X(1, 1, 2)	X(2, 1, 2)	X(1, 2, 2)	X(2, 2, 2)
X(1, 3, 2)	X(2, 3, 2)	X(1, 4, 2)	X(2, 4, 2)	X(1, 1, 3)	X(2, 1, 3)
X(1, 2, 3)	X(2, 2, 3)	X(1, 3, 3)	X(2, 3, 3)	X(1, 4, 3)	X(2, 4, 3)

٦ - ٣ افرض N ومجموعة متراسة خطية A بها N عنصراً مخزنة في الذاكرة . أكتب جزء برنامج فورتران يقوم بالتالي :

(أ) يحسب مجموع المربعات :

$$SUM = A_1^2 + A_2^2 + \dots + A_n^2$$

(ب) يحسب حاصل الضرب

$$PROD = (1 - A_1)(1 - A_2) \dots (1 - A_n)$$

(أ) أولاً اجعل SUM = 0.0 قبل إجراء التجميع :

```
SUM = 0.0
DO 100 K = 1, N
    SUM = SUM + A(K)**2
100 CONTINUE
```

(ب) أولاً اجعل PROD = 1.0 قبل إجراء الضرب . (هذا يشبه جعل SUM = 0.0 في (أ)) .

```
PROD = 1.0
DO 100 K = 1, N
    PROD = PROD*(1.0 - A(K))
100 CONTINUE
```

٦ - ٤ أوجد المخرج في البرنامج التالي :

```

INTEGER A(10)
DO 100 I = 1, 10, 2
    A(I) = 2*I - 3
    A(I + 1) = I**2 - 5
100 CONTINUE
DO 200 I = 2, 10, 3
    A(I) = A(I + 2) - A(I)
200 CONTINUE
WRITE(6, i0) A
10 FORMAT(1X, 5I8)
STOP.
END

```

تعلن الجملة الأولى أن A مجموعة متراصة خطية بها 10 عناصر أول حلقة DO تنفذ ما يلي :

(١) أولاً عندما $I = 1$ وهذا يعطى $A(1) = -1$ ، $A(2) = -4$
 (٢) ثم عندما $I = 3$ وهذا يعطى $A(3) = 3$ ، $A(4) = 4$
 (٣) ثم عندما $I = 5$ وهذا يعطى $A(5) = 7$ ، $A(6) = 20$
 (٤) ثم عندما $I = 7$ وهذا يعطى $A(7) = 11$ ، $A(8) = 44$
 (٥) أخيراً عندما $I = 9$ وهذا يعطى $A(9) = 15$ ، $A(10) = 76$
 وتنفذ حلقة DO الثانية ما يلي :

(١) أولاً $I = 2$ تغير قيمة $A(2)$ كما يلي :

$$A(2) \leftarrow A(4) - A(2) = 4 - (-4) = 8$$

(٢) ثم $I = 5$ تغير قيمة $A(5)$ كما يلي :

$$A(5) \leftarrow A(7) - A(5) = 11 - 7 = 4$$

(٣) أخيراً $I = 8$ تغير قيمة $A(8)$ كما يلي :

$$A(8) \leftarrow A(10) - A(8) = 76 - 44 = 32$$

يطبع زوج الجمل WRITE-FORMAT قيم A كل خمسة منها على السطر وحقول ذات عرض 8 حروف . من ثم سيظهر المخرج كما يلي :

-1	8	3	4	4
20	11	32	15	76

٦ - ٥ افترض N ومجموعة متراصة خطية A بها عدد N عنصر مخزنة في الذاكرة . أكتب جزءاً من برنامج فورتران يقوم بالآتي :

(أ) يبدل قيم A_1 و A_2 داخلياً يبدل قيم A_3 و A_4 داخلياً وهكذا ، (نفرض هنا أن N زوجية) .

(ب) يبدل دائرياً قيم A بحيث A_1 تحتوى القيمة الأصلية A_2 وتحتوى A_2 قيمة A_3 الأصلية ، وهكذا إلى أن تحتوى A_n قيمة A_1 الأصلية .

(أ) تذكر أولاً أننا فيدل قيم X و Y باستخدام مخزن مؤقت T كما يلي :

```
T = X
X = Y
Y = T
```

وبذلك يكون لدينا جزء البرنامج :

```
DO 100 K = 1, N, 2
  T = A(K)
  A(K) = A(K + 1)
  A(K + 1) = T
100 CONTINUE
```

(ب) أولاً ، نخزن A_1 في T مؤقتاً ، ثم ، نجعل $A(1) = A(2)$ و $A(2) = A(3) \dots A(N) = A(N-1)$ ، ثم أخيراً ، نجعل $A(N) = T$ وبذلك يكون لدينا :

```
T = A(1)
NN = N - 1
DO 100 K = 1, NN
  A(K) = A(K + 1)
100 CONTINUE
A(N) = T
```

لاحظ مرة أخرى أننا نستخدم الجملة $NN = N - 1$ حيث أننا لا نستطيع استخدام $N - 1$ كعامل .

برامج

٦ - ٦ إدرس مجموعة الأرقام x_1, x_2, \dots, x_n . يعرف كل من المتوسط الحسابي \bar{x} والتباين ، والانحراف المعياري σ بالآتي :

$$\bar{x} = (x_1 + x_2 + \dots + x_n) / n = \left(\sum_{k=1}^n x_k \right) / n$$

$$\text{variance} = \frac{\sum_{k=1}^n (x_k - \bar{x})^2}{n}$$

$$\sigma = \sqrt{\text{variance}}$$

أفرض عدد N من الأرقام مخزنة في مجموعة متراسة مخطة A . أكتب جزء برنامج فورتران يخزن المتوسط ، التباين والانحراف المعياري للأرقام في AVE و VAR و SD على الترتيب (قارن مع مسألة ٥ - ٩ حيث استخدمنا المعادلة الثانية للتباين) .

فيما يلي جزء البرنامج :

```
SUM1 = 0.0
DO 100 K = 1, N
  SUM1 = SUM1 + A(K)
100 CONTINUE
AVE = SUM1/FLOAT(N)
SUM2 = 0.0
DO 200 K = 1, N
  SUM2 = SUM2 + (A(K) - AVE)**2
200 CONTINUE
VAR = SUM2/FLOAT(N)
SD = SQRT(VAR)
```

٦ - ٧ إدارة سلسلة محلات لها 6 فروع ، ولكل فرع نفس الإدارات العشر . تخزن المبيعات الأسبوعية للسلسلة في مجموعة متراصة ، (6×10) باسم SALES . تشير SALES (I, J) هنا إلى المبيعات الأسبوعية في المحل رقم I والإدارة J . قم بكتابة جزء برنامج فورتران يقوم بالآتي :

- (أ) يطبع مجموع المبيعات الأسبوعية لكل فرع .
 (ب) يطبع مجموع المبيعات الأسبوعية لكل إدارة .
 (ج) يطبع مجموع المبيعات الأسبوعية للسلسلة بأكملها .

افترضنا هنا أن STORE و DEPT تم تعريفها بواسطة مجموعات مترابطة خطية باستخدام الجملة :

DIMENSION STORE(6), DEPT(10)

(أ) نجمع مبيعات فرع معين I ، بدءاً من $STORE(I) = 0$ في البداية .

```

WRITE(6, 10)
10 FORMAT('1', 5X, 'STORE', 5X, 'SALES'//)
DO 99 I = 1, 6
    STORE(I) = 0.0
    DO 88 J = 1, 10
        STORE(I) = STORE(I) + SALES(I, J)
88    CONTINUE
    WRITE(6, 20) I, STORE(I)
20    FORMAT(6X, I3, 5X, F9.2)
99    CONTINUE
  
```

(ب) نجمع مبيعات إدارة معينة I بدءاً من $DEPT(J) = 0$ في البداية .

```

WRITE(6, 30)
30 FORMAT('0', 2X, 'DEPARTMENT', 3X, 'SALES'//)
DO 77 J = 1, 10
    DEPT(J) = 0.0
    DO 66 I = 1, 6
        DEPT(J) = DEPT(J) + SALES(I, J)
66    CONTINUE
    WRITE(6, 40) J, DEPT(J)
40    FORMAT(3X, I5, 5X, F9.2)
77    CONTINUE
  
```

(ج) نستطيع أن نجمع القيم المخزنة في المجموعة المترابطة SALES أو في STORE أو في DEPT (أو في كل من الثلاثة) للتأكد من النتائج . نفعل ذلك أيضاً للمجموعة المترابطة STORE .

```

CHAIN = 0.0
DO 55 K = 1, 6
    CHAIN = CHAIN + STORE(K)
55    CONTINUE
WRITE(6, 50) CHAIN
50    FORMAT('0', 'TOTAL WEEKLY SALES'/6X, F9.2)
  
```

٦ - ٨ . إطبغ أول 50 حد من أرقام فيبوناتس بحيث يكون هناك خمسة أرقام في السطر .

تذكر أن أرقام فيبوناتس هي :

1, 1, 2, 3, 5, 8, 13, 21, ...

أى أن كلا من الحدين الأولين يساوى 1 ونحصل على كل حد بعد ذلك بجمع الحدين السابقين .

2 = 1 + 1, 3 = 1 + 2, 5 = 2 + 3, 8 = 3 + 5, ...

وبذلك ، نجعل :

$J(1) = 1, J(2) = 1, \text{ and } J(I) = J(I - 2) + J(I - 1)$

عندما $I > 2$. و

```

DIMENSION J(50)
J(1) = 1
J(2) = 1
DO 100 I = 3, 50
    J(I) = J(I - 2) + J(I - 1)
100 CONTINUE
WRITE(6, 10)
10 FORMAT('1', 15X, 'FIBONACCI SEQUENCE'//)
WRITE(6, 20) J
20 FORMAT(1X, 5(3X, I6))
STOP
END

```

لاحظ أن هناك خمسة مواصفات رقمية فقط في جملة FORMAT التي تحمل رقم السطر 20 وذلك بسبب طباعة كل خمسة أرقام على سطر .

٦ - ٩ (أ) افترض أن مجموعة متراصة خطية B بها M عنصر . أكتب جزء البرنامج لإيجاد المكان J بحيث تحتوي B(J) على أكبر قيمة مطلقة مخزنة في B (أنظر مثال ٦ - ١) .

(ب) افترض أن A مصفوفة (M × N) مخزنة في الذاكرة فعلا . اكتب جزء برنامج فورتران يقوم بالتالي :

(١) يجد المكان J بحيث تحتوي J على أكبر قيمة مطلقة في العمود الأول .

(٢) يبدل الصف الأول مع الصف رقم J .

(من ثم ، ستمتوى A(1, 1) على القيمة الكبرى المطلقة في العمود الأول) .

(أ) نجعل $J = 1$ و $BIG = ABS(B(1))$ ثم نقارن BIG مع بقية عناصر B إذا كانت BIG أقل من $ABS(B(K))$ ، فنجعل $BIG = ABS(B(K))$ ونغير J إلى K . وفيما يلي البرنامج :

```

J = 1
BIG = ABS(B(1))
DO 100 K = 1, M
    IF(BIG.GE.ABS(B(K))) go to 100
    BIG = ABS(B(K))
    J = K
100 CONTINUE
    
```

يحمل الدليل K في البداية مساوياً 1 وتأخذ في الاعتبار أن M يمكن أن تساوى 1 .

(ب) جزء البرنامج به إجراءان (١) إيجاد قيمة J (٢) إذا كانت 1 بخارج J نبديل الصف الأول بالصف رقم J .

```

C
C FIND THE POSITION J WITH LARGEST ABSOLUTE VALUE
C
    
```

```

J = 1
BIG = ABS(A(1,1))
DO 100 K = 1, M
    IF(BIG.GE.ABS(A(K,1)))GO TO 100
    BIG = ABS(A(K,1))
    J = K
    
```

```

100 CONTINUE
IF(J.EQ.1) GO TO 500
    
```

```

C
C INTERCHANGE THE FIRST AND JTH ROWS
C
    
```

```

DO 200 L = 1, N
    TEMP = A(1, L)
    A(1, L) = A(J, L)
    A(J, L) = TEMP
    
```

```

200 CONTINUE
500 *****
    
```

لاحظ التشابه بين الجزء الأول من البرنامج والجزء (أ) .

٦ - ١٠ اكتب جزء البرنامج الذي يضيف عنصر D في المكان K من المجموعة المتراسة A(1) و A(2) و ... و A(N) .

قبل أن تخصص D إلى A(K) يجب أن تحرك جزء المجموعة المتراسة A(K) و A(K + 1) و ... و A(N) إلى أسفل مكان واحد . يتم هذا بالبداية من نهاية المجموعة أي ، بأن تخصص A(N) إلى A(N + 1) ثم A(N - 1) إلى A(N) وهكذا حتى A(K) إلى A(K + 1) (أنظر قسم ٦ - ٨) لاحظ أولاً أننا يجب أن نحرك N - K + 1 عنصراً ، وليس N - K حيث أننا نحرك أيضاً A(K) . نستخدم حلقة DO بدليل J = 1, 2, ..., N - K + 1 . فيما يلي العلاقة بين الدليل J والأجراء الذي يتخذ :

قيمة J	الأجراء
1	A(N + 1) ← A(N)
2	A(N) ← A(N - 1)
3	A(N - 1) ← A(N - 2)
...
N - K + 1	A(K + 1) ← A(K)

فيما يلي جزء البرنامج :

```

NN = N - K + 1
DO 100 J = 1, NN
    A(N - J + 2) = A(N - J + 1)
100 CONTINUE
A(K) = D

```

أو ببساطة

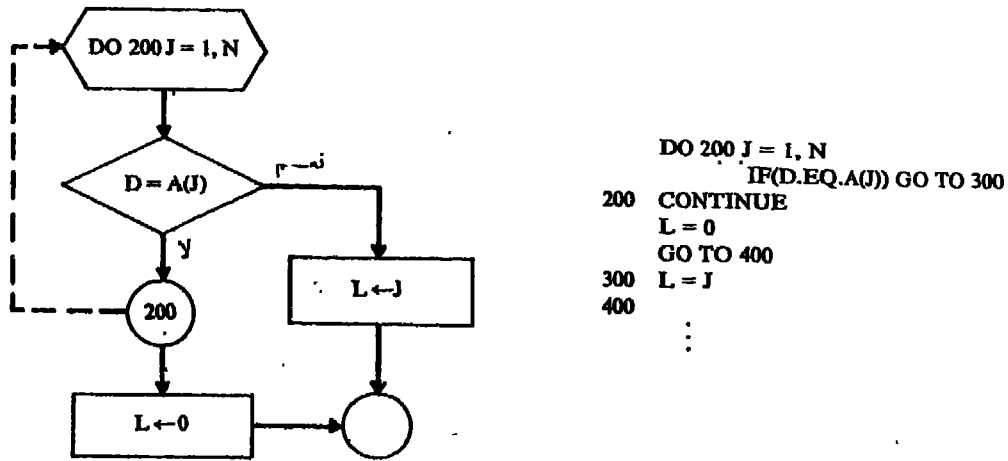
```

NN = N - K + 1
DO 100 J = 1, NN
    L = N - J + 1
    A(L + 1) = A(L)
100 CONTINUE
A(K) = D

```

١١ - ٦ (البحث الخطي) إذا أعطيت مجموعة متراسة $A(1)$ و $A(2)$ و \dots و $A(N)$ وعنصر D ، أكتب جزء البرنامج الذي يجد المكان L للعنصر D عندما تقع قيمة D في المجموعة المتراسة ، أو يجعل $L = 0$ عندما لا تقع قيمة D في المجموعة المتراسة .

لا توجد أي معلومات أخرى معطاة عن المجموعة المتراسة ، والطريقة الأكثر تلقائية لحل المسألة هي أن نقارن D مع كل عنصر من A أي ، نختبر إذا كانت $D = A(1)$ ثم نختبر إذا كانت $D = A(2)$ وهكذا . وتسمى هذه الطريقة البحث الخطي أو البحث المتعاقب (المتسلسل حيث أنها تبحث عن D باختبارها مع قيم عناصر A واحدة تلو الأخرى) (تناقش طرق أخرى للبحث في الفصل الثامن) . واضح أن هذه الطريقة تحتاج إلى عدد N من المقارنات لإيجاد D تحتاج عدد N من المقارنات عندما لا تكون D في المجموعة المتراسة . يعرض الشكل ٦ - ١٠ خريطة سير العمليات للنظام الحسابي (الخوارزم) وما يقابلها بالفورتران .



شكل ٦ - ١٠

١٢ - ٦ (الفرز) . يقال أن المجموعة المتراسة مفروزة إذا كانت عناصرها مرتبة بأى نظام . أكتب جزء البرنامج الذي يفرز المجموعة المتراسة $A(1)$ ، $A(2)$ ، \dots ، $A(N)$ بحيث تكون عناصرها مرتبة تصاعدياً .

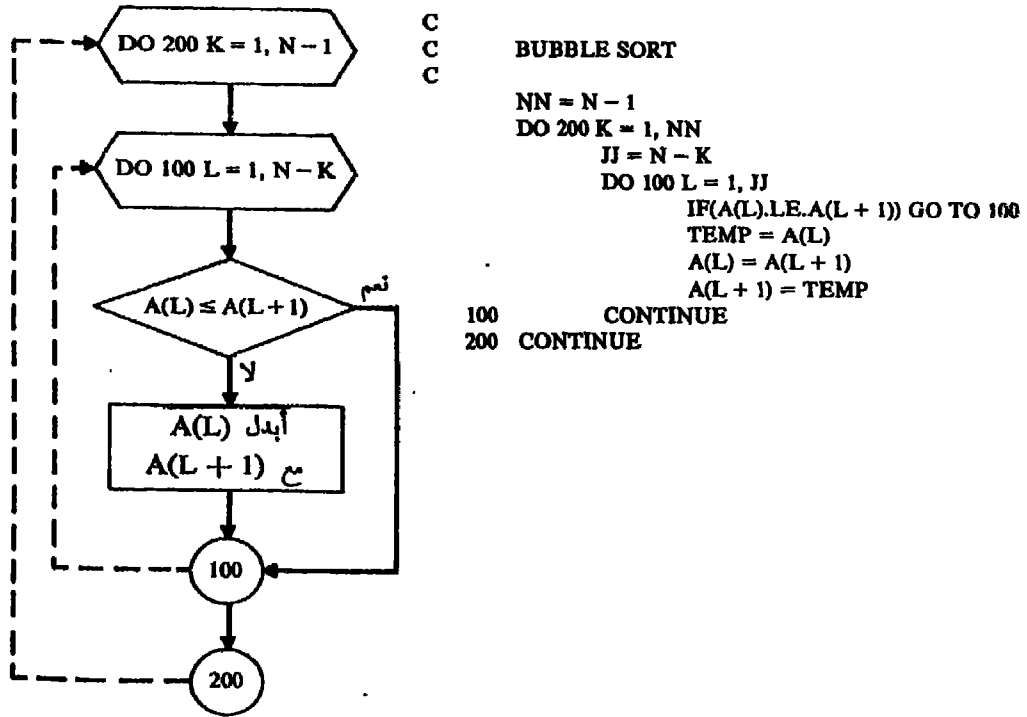
تفرز A بالنظام الحسابي (الخوارزم) المعروف بـ « الفرز الفقاعي » وسوف (تناقش طرق أخرى للفرز في الفصل الثامن) . نقارن أولاً $A(1)$ و $A(2)$ ونرتبهما في الترتيب المطلوب ، أي بحيث يكون $A(1) < A(2)$ ثم نقارن $A(2)$ و $A(3)$ ونرتبهما بحيث يكون $A(2) < A(3)$ ثم نقارن $A(3)$ و $A(4)$ ونرتبهما بحيث يكون $A(3) < A(4)$. يكل هذا إلى $A(N-1) < A(N)$ - وبيناً تلق نظرة شاملة على كل العناصر « يطفو » العنصر الأكبر إلى المكان N .

تكرر العملية السابقة العناصر $A(1)$ ، $A(2)$ ، ... ، $A(N-1)$ بعدما تلقى نظرة شاملة على هذا الناصر ، يطفو العنصر الأكبر الثاني إلى الموقع $N-1$ وهكذا بعد عدد $N-1$ من المرات ستكون المجموعة المترابطة A مبروزة في الترتيب التصاعدي .

ونذكر هنا شكل النظام الحسابي (الخوارزم) :

- الخطوة ١ : لكل $L = 1, 2, \dots, N-1$ قارن $A(L)$ مع $A(L+1)$ ورتبهما بحيث تكون $A(L) < A(L+1)$
- الخطوة ٢ : لكل $L = 1, 2, \dots, N-2$ قارن $A(L)$ مع $A(L+1)$ ورتبهما بحيث تكون $A(L) < A(L+1)$
-
- الخطوة $N-1$: بالنسبة لـ $L = 1$ قارن $A(L)$ مع $A(L+1)$ ورتبهما بحيث يكون $A(L) < A(L+1)$.

لاحظ أن الإجراء الذي يتخذ متشابه في كل خطوة . من ثم ، يمكن أن يكون لدينا حلقة DO محكمة بدائل K يأخذ القيم 1 ، 2 ، ... ، $N-1$ (تيمناً لعدد $N-1$ خطوة) ، وبداخل هذه الحلقة حلقة أخرى داخلية بدليل L يأخذ القيم 1 ، 2 ، ... ، $N-K$. يبين شكل ١١-٦ خريطة سير العمليات لهذا النظام الحسابي وترجمته للفورتراد



شكل ١١-٦

٦- ١٣ أكتب برنامجاً يقبل 25 رقماً صحيحاً موجباً مختلفاً N_1, N_2, \dots, N_{25} ويطبع ، على أسطر منفصلة ، كل زوج من الأرقام الصحيحة التي يكون مجموعها 75 . (ملحوظة : إذا كان 25 و 50 رقمين صحيحين من ثم فإننا نريد أما أن نطبع 25 و 50 أو نطبع 50 ، 25 ولكن ليس الإثنين معاً) .

$$N_1 + N_k = 75$$

أولا نقارن N_1 بكل من N_2, N_3, \dots, N_{25} لئلا نرى إذا كانت

$$N_2 + N_K = 75$$

حيث $K > 1$ ثم نقارن N_2 بكل من N_3, N_4, \dots, N_{24} لئلا نرى إذا كانت

$$N_J + N_K = 75$$

حيث $K > 2$ وهكذا . وبمعنى آخر لقيم J من 1 ، 2 ، ... ، 24 نتحقق لئلا نرى إذا كانت

$$N_J + N_K = 75$$

$J > K$. ويمكن عمل هذا بحلقات DO المتداخلة كما يلي :

```

DIMENSION N(25)
READ(5, 10) N
10  FORMAT(5I5)
DO 100 J = 1, 24
    JJ = J + 1
    DO 200 K = JJ, 25
        IF (N(J) + N(K).NE.75) GO TO 200
        WRITE(6, 20) N(J), N(K)
        FORMAT(1X, I5, 3X, I5)
    20  CONTINUE
200  CONTINUE
100  CONTINUE
STOP
END

```

٦ - ١٤ أدرس المادة متعددة الحلود :

$$y = 2x^4 - 5x^3 + 6x^2 - 8x + 9$$

(أ) اكتب جزء برنامج فورتران يخزن ويطيح قيمة y لقيم x حيث: $10, -9, \dots, -10, x = -10$ ، أي ، لقيم x الصحيحة من -10 إلى 10

(ب) اكتب جزء برنامج فورتران يخزن ويطيح قيمة y لقيم x حيث : $N, -N+1, \dots, -N, x = ,$ حيث يقرأ N كبيان . (افرض $0 < N \leq 25$)

(أ) لاحظ أن هناك 21 قيمة لـ X من ثم نستخدم حلقة DO بدليل $K: K = 1, 2, \dots, 21$ وفيما يلي العلاقة بين X و K

$$\text{قيمة } X : -10, -9, -8, \dots, 9, 10$$

$$\text{قيمة } K : 1, 2, 3, \dots, 20, 21$$

لاحظ أن : $X = -11 + K$. فيما يلي جزء البرنامج :

```

DIMENSION Y(21)
DO 100 K = 1, 21
    X = -11 + K
    Y(K) = (((2.*X - 5.)*X + 6.)*X - 8.)*X + 9.
    WRITE(6, 10) X, Y(K)
    FORMAT(1X, F5.1, 2X, F8.1)
10  CONTINUE
100 CONTINUE

```

(ب) لدينا الآن $2N + 1$ قيمة لـ X (لاحظ أن $2N + 1 \leq 51$ حيث $N \leq 25$) من ثم نستخدم حلقة DO بدليل K حيث : $K = 1, 2, \dots, 2N + 1$. هنا X و K . بينهما العلاقة $X = -N - 1 + K$.

```

DIMENSION Y(51)
READ(5, 20) N
20  FORMAT(I5)
    NN = 2*N + 1
    DO 88 K = 1, NN
        X = -N - 1 + K
        Y(K) = (((2.*X - 5.)*X + 6.)*X - 8.)*X + 9.
        WRITE(6, 30) X, Y(K)
30  FORMAT(1X, F5.1, 2X, F8.1)
88  CONTINUE

```

لاحظ أننا استخدمنا علاقة هورنر لحساب Y (أنظر قسم ٦ - ٨) .

٦ - ١٥ (التوزيع التكراري) مجموعة بها 200 بطاقة تحتوي على رقم صحيح موجب K في كل بطاقة ، حيث $K \leq 25$ اكتب البرنامج الذي يطبع الأرقام الصحيحة الموجبة من 1 إلى 25 وعدد المرات التي يظهر فيها الرقم الصحيح في المجموعة .

نستخدم النظام الحسابي (الخوارزم) الآتي . تشير إلى أن N مجموعة متراصة خطية بها 25 عنصراً ، ونجمل $N(K)$ تشير إلى عدد المرات التي ظهرت فيها قيمة K . بالتحديد ، نجمل في البداية كل $N(K) = 0$ وكلما ظهرت قيمة K تزيد $N(K)$ بواحد . على سبيل المثال . نجمل في البداية $N(8) = 0$ فإذا ظهرت القيمة $K = 8$ وتكررت 14 مرة تصبح $N(8) = 14$ بعد قراءة كل البطاقات . وفيما يلي البرنامج :

```

DIMENSION N(25)
C
C  SET EACH N(K) = 0
C
    DO 100 K = 1, 25
        N(K) = 0
100  CONTINUE
C
C  READ AND COUNT THE NUMBERS
C
    DO 200 J = 1, 200
        READ(5, 10) K
        FORMAT(I5)
        N(K) = N(K) + 1
200  CONTINUE
C
C  PRINT THE DISTRIBUTION
C
    DO 300 K = 1, 25
        WRITE(6, 20) K, N(K)
        FORMAT(6X, 2(I4, 5X))
300  CONTINUE
    STOP
    END

```

٦ - ١٦ اعتبر A مجموعة متراصة خطية بها N عنصر . نحصل على أول فرق DA للمجموعة المتراصة A بطرح كل عنصر ، من العنصر التالي له في المجموعة المتراصة ماعداً العنصر الأخير

$$DA(K) = A(K + 1) - A(K)$$

حيث $1 \leq K \leq N-1$ لاحظ أن DA به $N-1$ عنصر . الفرق الثاني للمجموعة المتراسة $D2A$ يعرف كالفرق الأول DA وهكذا .

افترض أن A مخزنة في الذاكرة . أكتب جزء البرنامج الذي يجد أول فرق DA ، ثاني فرق $D2A$ وثالث فرق $D3A$ لمتتالية A :

2, 8, -3, 5, 9, -4, 8, 0, -8, 16

ثم يطبع A ، DA ، $D2A$ ، $D3A$.

يمكن أن تبين الفروض كما يلي :

2	8	-3	5	9	-4	8	0	-8	16
6	-11	8	4	-13	12	-8	-8	24	
-17	19	-4	-17	25	-20	0	32		
	36	-23	-13	42	-45	20	32		

أي ، الصف الثاني من الأرقام هو الفرق الأول ، والصف الثالث هو الفرق الثاني والصف الأخير هو الفرق الثالث . وفيما يلي البرنامج .

```

C
C   FIND THE FIRST DIFFERENCE
C
      N = N - 1
      DO 100 K = 1, N
          DA(K) = A(K + 1) - A(K)
100  CONTINUE
C
C   FIND THE SECOND DIFFERENCE
C
      N = N - 1
      DO 200 K = 1, N
          D2A(K) = DA(K + 1) - DA(K)
200  CONTINUE
C
C   FIND THE THIRD DIFFERENCE
C
      N = N - 1
      DO 300 K = 1, N
          D3A(K) = D2A(K + 1) - D2A(K)
300  CONTINUE
C
C   PRINT OUT THE ARRAYS
C
      WRITE(6, 10) A
      WRITE(6, 10) DA
      WRITE(6, 10) D2A
      WRITE(6, 10) D3A
10   FORMAT(1X, 10(2X, F8.2))

```

إن لم تكن المجموعات المتراسة DA و $D2A$ و $D3A$ مطلوبة فيما بعد في البرنامج ، يمكن أن نستخدم حلقات DO التكرارية المتداخلة .

حلقات DO التكرارية الضمنية

٦ - ١٧ أعد كتابة كل جملة في حلقة DO الضمنية بحيث لا يكون هناك أخطاء

READ(5, 10), (A(K), K = 1, 20) (أ)
 WRITE(6, 20), A(I, I), S, B(I), I = 1, N, 3 (ب)
 READ(5, 30) (A(J), J = 1, 7), B, (C(K), K = 1, M, L) (ج)
 WRITE(6, 40) (A(L, K + 3), K = 1, 7), L = 1, 5) (د)
 READ(5, 10) (A(K), K = 1, 20) (أ)
 WRITE(6, 20) ((A(I, I), S, B(I)), I = 1, N, 3) (ب)

(ح) لا يوجد أخطاء .

(د) لا نستطيع أن نستخدم $K + 3$ كدليل ، ولكن نستطيع أن نكتب

$KK = K + 3$
 WRITE(6, 40) ((A(L, KK), KK = 4, 10), L = 1, 5)

٦ - ١٨ صف المخرج إذا نفذنا كل ثنائي من الجمل WRITE-FORMAT التالية :

WRITE(6, 10) (A(K), K = 2, 25, 3) (أ)
 10 FORMAT(1X, 5(3X, F10.2))
 WRITE(6, 10) (A(L), B, C(3, L), L = 1, 8, 2) (ب)
 10 FORMAT(1X, 5(3X, F10.2))
 WRITE(6, 10) (B(J), J = 1, 4), (B(K), K = 3, 10, 4) (ح)
 10 FORMAT(1X, 5(3X, F10.2))

(أ) لاحظ أن 2 هي القيمة الابتدائية ل K وأن 3 هي معامل الزيادة و 25 قيمة النهاية. من ثم تطبع كل خمسة أرقام من A(2) و A(5) و A(8) و A(11) و A(14) و A(17) و A(20) و A(23) على سطر منفصل وبذلك يظهر المخرج كما يلي :

A(2) A(5) A(8) A(11) A(14)
 A(17) A(20) A(23)

(ب) لاحظ أن الحاسب ينفذ حلقة DO الضمنية أولاً عندما $L=1$ ثم عندما $L=3$ ثم عندما $L=5$ وأخيراً عندما $L=7$. ومن ثم A(1) و B و C(3,1) و A(3) و B و C(3,3) و A(5) و B و C(3,5) و A(7) و C(3,7) تطبع كل خمسة أرقام منها على سطر منفصل :

A(1) B C(3, 1) A(3) B
 C(3, 3) A(5) B C(3, 5) A(7)
 B C(3, 7)
 B(1) B(2) B(3) B(4) B(3) (ح)
 B(7)

٦ - ١٩ صف أخرج عند تنفيذ ما يلي :

```
WRITE(6, 20) ((A(J, K), J = 2, 8, 3), K = 1, 4)
20 FORMAT(IX, 5(3X, F10.2))
```

تمتلح حلقة DO الضمنية الداخلية : $A(2, K)$ و $A(5, K)$ و $A(8, K)$ ولما كانت $K = 1, 2, 3, 4$ فإننا نحصل على 12 قيمة تطبع كل خمسة منها على سطر . ويظهر الأخرج بالصورة التالية :

```
A(2, 1)  A(5, 1)  A(8, 1)  A(2, 2)  A(5, 2)
A(8, 2)  A(2, 3)  A(5, 3)  A(8, 3)  A(2, 4)
A(5, 4)  A(8, 4)
```

٦ - ٢٠ مصفوفة $A(N \times N)$ مخزنة في الذاكرة . أكتب جزء برنامج فورتران يطبع عناصر A الموجودة فوق القطر ، أي العناصر $A(I, J)$ لكل $I < J$. نريد أن تطبع العناصر :

```
A12 A13 A14 ... A1N
      A23 A24 ... A2N
      .....
      AN-1,N
```

يشير الدليل الأول I إلى صف المتغير ، ومن ثم يشير من 1 إلى $N-1$. ويشير الدليل الثاني إلى العمود ، ومن ثم يشير من $I+1$ إلى N . لا نستطيع أن نستخدم حلقة DO الضمنية :

```
WRITE(6, 10) ((A(I, J), J = I + 1, N), I = 1, N - 1)
```

حيث لا يمكن أن تكون $I + 1$ ، $N - 1$ أدلة . ومع ذلك نستطيع أن نستخدم حلقة DO التكرارية مع حلقة DO التكرارية الضمنية :

```
NN = N - 1
DO 100 I = 1, NN
      K = I + 1
      WRITE(6, 10) (A(I, J), J = K, N)
10    FORMAT(1X, 8(3X, F10.2))
100 CONTINUE
```

مسائل تكميلية

المجموعات المترابطة

٦ - ٢١ اكتشف الأخطاء ، إن وجدت ، في كل جملة من جمل DIMENSION التالية :

- DIMENSION, J(8), INTEREST(5, 10), (أ)
- DIMENSION A(4, 0, 3), B(6) C(9, 4) (ب)
- DIMENSION, A(M, N) XYZ(4, -8, 7) (ج)

٦ - ٢٢ حدد عدد العناصر في المجموعات المتراسة NAME و ID و X و Y تبعاً للجملة التالية

DIMENSION NAME(5, 300), ID(3, 5, 2), X(50), Y(4, 8)

٦ - ٢٣ باستخدام المجموعة المتراسة ID في المسألة السابقة ٦ - ٢٢ ، صف المخرج إذا نفذنا الزوج التالي من الجمل :

```
WRITE(6, 10) ID
10 FORMAT(1X, 8I8)
```

٦ - ٢٤ باستخدام المجموعة المتراسة Y في المسألة ٦ - ٢٢ ، أكتب زوج الجمل WRITE-FORMAT الذى يطبع الصفوف الأربعة من Y على السطور الأربعة الأولى من صفحة الطباعة على الترتيب .

٦ - ٢٥ أوجد المخرج الخاص بكل جزء برنامج ما يلى :

<pre>INTEGER X(6) DO 100 K = 1, 6, 2 X(K) = 3*K X(K + 1) = K + 2 100 CONTINUE DO 200 J = 1, 6, 3 X(J) = X(J) + X(J + 1) 200 CONTINUE IF(X(2).LT.7) X(2) = X(3) WRITE(6, 20) X 20 FORMAT(1X, 6I8).</pre>	(ب)	<pre>DIMENSION ID(5) DO 100 K = 1, 5 ID(K) = 2*K 100 CONTINUE ID(2) = ID(2) + ID(3) ID(5) = ID(5) - ID(4) WRITE(6, 10) ID 10 FORMAT(1X, 5I8)</pre>	(أ)
---	-----	--	-----

٦ - ٢٦ افترض مجموعة متراسة خطية A بها N عنصر مخزنة في الذاكرة . أكتب جزء البرنامج الذى يقوم بالتالى :

(أ) يبدل A_1 و A_2 ما إذا كانت A_1 أكبر من A_2 فقط ، يبدل A_3 و A_4 ما إذا كانت A_3 أكبر من A_4 فقط ، وهكذا (نفترض هنا N زوجية)

(ب) يبدل دائرياً قيم A بحيث A_2 تحتوى القيمة الأصلية A_1 و A_3 تحتوى القيمة الأصلية لـ A_2 وهكذا ، وتحتوى A_N القيمة الأصلية لـ A_N .

حلقات DO التكرارية الضمنية

٦ - ٢٧ صف المخرج إذا نفذنا زوج الجمل WRITE-FORMAT التالى :

<pre>WRITE(6, 10) (A(N), N = 4, 28, 5) 10 FORMAT(1X, 3F15.2)</pre>	(أ)
<pre>WRITE(6, 20) (J(L), K, N(3, L), L = 2, 13, 3) 20 FORMAT(1X, 4I10)</pre>	(ب)

٦ - ٢٨ صف المخرج إذا نفذنا زوج الجمل WRITE-FORMAT التالي :

- (أ) $WRITE(6, 30) (A(M), M = 1, 5), X, (B(N), N = 1, 7, 3)$
 30 $FORMAT(1X, 4F15.2)$
- (ب) $WRITE(6, 40) ((A(M, N), M = 1, 5), N = 1, 7, 3)$
 40 $FORMAT(1X, 4F15.2)$
- (ج) $WRITE(6, 50) ((A(M, N), N = 1, 7, 3), M = 1, 5)$
 50 $FORMAT(1X, 4F15.2)$

برامج

٦ - ٢٩ أدى فصل به 35 طالباً إمتحان تراوح درجاته ما بين 0 و 100 أكتب برنامج فورتران يجد : (أ) متوسط الدرجات ، (ب) عدد الطلبة الراسبين أى الحاصلين على درجات أقل من 60 درجة و (ج) عدد الطلبة الممتازين أى ، الحاصلين على 100 درجة .

٦ - ٣٠ يوزع لطلاب سبعة اختبارات (تراوح درجاتها ما بين 0 و 100) ، ودرجته النهائية هي متوسط أعلى ست درجات في الاختبارات . باعتبار أن الدرجات السبع للاختبار قد تم تثقيبها على بطاقة بيانات . اكتب جزء برنامج الفورتران الذي يجد درجة الطالب النهائية . (تلميح . اطرح أصغر درجة من مجموع الدرجات السبع SUM) .

٦ - ٣١ يستثمر رجل رأس ماله في حساب توفير يربح 7 في المائة كل سنة ولمدة 10 سنوات بغائدة مركبة . اكتب البرنامج الذي يطبع ما وصل إليه حسابه كل سنة لمدة 20 سنة (لاحظ أنه ليست هناك استثمارات إضافية في آخر 10 سنوات)

٦ - ٣٢ سلسلة إدارة فروع بها 6 محلات ، كل فرع به نفس عدد الإدارات وهي 12 . المبيعات الأسبوعية للسلسلة تخزن في مجموعة متراسة $(7 \times 12 \times 6)$ باسم SALES بحيث تشير SALES(I, J, K) إلى المبيعات في الفرع I الإدارة J في اليوم K . اكتب جزء برنامج فورتران يقوم بالتالي :

(أ) يطبع مجموع المبيعات الأسبوعية لكل فرع .
 (ب) يطبع مجموع المبيعات الأسبوعية لكل إدارة .
 (ج) يطبع مجموع المبيعات الأسبوعية للسلسلة بأكملها .
 (د) قارن مع مسألة (٦ - ٧) .

٦ - ٣٣ سلسلة إدارة محلات بها 14 فرع (مترقة من 1 إلى 14) وكل فرع به نفس عدد الإدارات وهي 8 إدارات . يقدم كل فرع بطاقة بيانات كل أسبوع إلى المكتب الرئيسي مثقب عليها رقم الفرع في الأعمدة 1 إلى 3 ثم تثقب 8 أرقام تمثل المبيعات الأسبوعية في كل إدارة . اكتب جزء برنامج فورتران يقوم بالتالي :

(أ) يقرأ البيانات في المجموعة المتراسة SALES بحيث تحوى SALES(I, J) على المبيعات الأسبوعية في المحل I و الإدارة J . (افرض أن بطاقات البيانات ليست مرتبة بأي ترتيب معين) .
 (ب) يطبع مجموع المبيعات لكل فرع .
 (ج) يطبع رقم الفرع صاحب أكبر مبيعات .

٦ - ٣٤ إذا أعطيت أرقاماً صحيحة موجبة I و J وكانت أرقام فيوناسي المعممة لـ $K > 2$ تعرف بما يلي :

$$JFIB(1) = I, \quad JFIB(2) = J, \quad \text{and} \quad JFIB(K) = JFIB(K-2) + JFIB(K-1)$$

(أنظر مسألة (٦ - ٨)) . اكتب البرنامج الذي يقرأ I و J ويطبع أول 30 حد من السلسلة بحيث يطبع ثلاثة أرقام في السطر .

٦ - ٣٥ اكتب برنامج الفورتان الذي :

- (أ) يقبل 25 رقماً صحيحاً موجباً ويقرر ما إذا كان أي رقمين من الأرقام الصحيحة مجموعها 15 .
 (ب) يقبل 15 رقماً صحيحاً موجباً ويحدد الرقم التالي لأصغر رقم والرقم التالي لأكبر رقم .
 (ج) يقبل 25 رقماً صحيحاً موجباً ويحدد أكبر رقم زوجي ، أن لم يجده يطبع THERE ARE NO EVEN INTEGERS
 (د) يقبل 25 رقماً صحيحاً موجباً ويحسب عدد الأرقام الزوجية منها ، وكذا عدد الأرقام الفردية .

٦ - ٣٦ ثقتب درجات لجنة اختيار دخول كلية لعدد 450 طالباً على بطاقات ، درجة واحدة في كل بطاقة . (تراوح تلك الدرجات ما بين 200 و 800) . اكتب البرنامج الذي يحدد عدد الدرجات بين 200 و 300 وبين 300 و 400 وهكذا .

٦ - ٣٧ اكتب البرنامج الذي يقرأ المعاملات a_1, a_2, \dots, a_{n-1} لكثير الحدود .

$$p(x) = a_1x^n + a_2x^{n+1} + \dots + a_nx + a_{n+1}$$

ويخزنها في المصفوفة الخطية COEF ويطبع قيم $p(x)$ لقيم $x = -5$ و -4.5 و -4 و 4.5 و 5 أي لقيم x من -5 إلى 5 بخطوات مقدارها 0.5 . استخدم طريقة هورنر لإيجاد قيمة $p(x)$ أي :

$$p(x) = ((\dots((a_1x + a_2)x + a_3) + \dots + a_{n-1})x + a_n)x + a_{n+1}$$

٦ - ٣٨ (نظرية المجموعات) افترض JSET و KSET مجموعتين متراسين خطية كل منهما يحتوي على أرقام صحيحة موجبة مختلفة عن بعضها .

- (أ) اكتب جزء البرنامج الذي يخزن في INTER العناصر المشتركة بين JSET و KSET أي تلك العناصر الموجودة في JSET و KSET أيضاً .
 (ب) اكتب جزء البرنامج الذي يخزن في JUNION العناصر كلها في JSET و KSET أي تلك العناصر في JSET أو KSET (أو كليهما) .

٦ - ٣٩ قائمة جرد البضائع في مستودع لأجهزة TV مخزنة في مجموعتين متراسيتين صحيحتين ID و AMOUNT كما يلي . يعطى كل مدخل في ID رقم طراز TV (رقم من ست خانانات) ، ويعطى المدخل المقابل في AMOUNT عدد التليفزيونات من هذا الطراز في المستودع . يقدم كل أمر شحن للوارد أو الصادر مجموعة من بطاقات البيانات (لها بطاقة خلفية) ثقتب عليها رقم الطراز في الأعمدة من 1 إلى 8 وثقتب كمية الصنف المشحون في الأعمدة من 11 إلى 20 . (حيث يكون العدد موجباً إذا كانت الشحنة واردة إلى المستودع ويكون سالباً إذا كانت صادرة من المستودع) . اكتب البرنامج الذي يعدل قائمة الجرد بعد كل أمر شحن . يجب أن يطبع أيضاً :

INSUFFICIENT QUANTITY MODEL NUMBER XXXXXX

إذا كان الأمر الصادر من المستودع لصنف يتجاوز العدد في الرصيد .

٦ - ٤٠ A مصفوفة $(N \times N)$ مخزنة في الذاكرة .

- (أ) اكتب البرنامج الذي يجمع العناصر فوق القطر الرئيسي ، أي العناصر $A(I, J)$ بحيث $I < J$.
 (ب) اكتب البرنامج الذي يجمع العناصر أسفل القطر الرئيسي ، أي ، العناصر $A(I, J)$ بحيث $I > J$.

٦ - ٤١ تحتوي (لعبة التصويب بالكرة) على مجموعة متراسة خطية A لعدد الأوتاد المضروبة في كل ضربة . أكتب جز ' برنامج الذى يزن النتيجة بعد كل دورة في مجموعة متراسة B وعلى ذلك B(10) تعطى النتيجة النهائية . اختبر البرنامج بابيانات التالية :

(أ) 4, 5, 10, 10, 6, 4, 7, 3, 8, 0, 7, 3, 10, 8, 1, 10, 7, 2
(ب) 6, 4, 7, 2, 10, 7, 3, 10, 5, 4, 3, 0, 10, 10, 8, 2, 7

٦ - ٤٢ (طريقة سيف) اكتب البرنامج الذى يطبع الأعداد الأولية التى أقل من 400 كما يلي :

- (أ) إجمل A مجموعة متراسة خطية بها 400 عنصر .
(ب) خزن K في A(K) أى ، إجمل $A(1) = 1$ و $A(2) = 2$ و $A(400) = 400 \dots$
(ج) اطبع A(2) ثم ضع كل مضاعفات 2 مساوية للصفر .
(د) اطبع العنصر التالي والذي لا يساوى صفرأ في المجموعة المتراسة وهو A(3) ، ثم ضع كل مضاعفات 3 مساوية للصفر .
(هـ) كرر العمليات السابقة طالما أن $A(K) \leq 20 = \sqrt{400}$
(ز) عند : $A(K) > 20$ اطبع كل العناصر غير الصفرية في A .

اجابات للمسائل التكميلية المختارة

٦ - ٢١ (أ) يجب ألا تكون هناك فصلة بعد DIMENSION ولا في النهاية .
(ب) يجب أن تكون هناك فصلة بعد B(6) . أيضاً لا يمكن أن يكون الصفر دليلاً .
(ج) يجب ألا تكون هناك فصلة بعد DIMENSION ؛ يجب أن تكون هناك فصلة قبل XYZ ؛ ولا يمكن أن تكون M و N و 8 — أدلة .

٦ - ٢٢ 1500 و 30 و 50 و 32 .

٦ - ٢٣ حيث أن اسم المجموعة المتراسة هو الذى يظهر فقط ، ستطبع ID بترتيب التخزين أى ستطبع بطريقة الأعمدة من الصفحة الأولى ثم بطريقة الأعمدة من الصفحة الثانية مع كل ثمانية عناصر على سطر .

٦ - ٢٤
WRITE(6, 20) ((Y(I, J), J = 1, 8), I = 1, 4)
20 FORMAT('1', 8F15.2/1X, 8F15.2/1X, 8F15.2/1X, 8F15.2)

٦ - ٢٥ (أ) الأرقام الصحيحة 2 و 10 و 6 و 8 و 2 ستطبع مضبوطة من الطرف الأيمن في حقول بعرض 8 خانات .
(ب) الأرقام الصحيحة 6 و 9 و 9 و 20 و 15 و 7 ستطبع في حقول بعرض 8 خانات .

(أ)
T = A(N) DO 99 K = 1, N, 2
NN = N - 1 IF(A(K).LE.A(K + 1)) GO TO 99
DO 88 K = 1, NN T = A(K)
A(N - K + 1) = A(N - K) A(K) = A(K + 1)
88 CONTINUE A(K + 1) = T
A(1) = T 99 CONTINUE

٦ - ٢٧ (أ) $A(4)$ و $A(9)$ و $A(14)$ على سطر واحد و $A(19)$ و $A(24)$ على السطر الثاني .
 (ب) $J(2)$ و K و $N(3, 2)$ و $J(5)$ على سطر واحد ؛ K و $N(3, 5)$ و $J(8)$ و K على السطر التالي ، و $N(1, 3)$ و $J(11)$ و K و $N(3, 11)$ على السطر الثالث .

٦ - ٢٨ (أ) $A(1)$ و $A(2)$ و $A(3)$ و $A(4)$ على سطر واحد ؛ $A(5)$ و X و $B(1)$ و $E(4)$ على السطر التالي و $B(7)$ على السطر الثالث .
 (ب) $A(1, 1)$ و $A(2, 1)$ و $A(3, 1)$ و $A(4, 1)$ على سطر واحد و $A(5, 1)$ و $A(1, 4)$ و $A(2, 4)$ و $A(3, 4)$ على السطر التالي و $A(1, 7)$ و $A(2, 7)$ و $A(4, 4)$ و $A(5, 4)$ على السطر الثالث ؛ $A(3, 7)$ و $A(4, 7)$ و $A(5, 7)$ على السطر الرابع .
 (ج) $A(1, 1)$ و $A(1, 4)$ و $A(1, 7)$ و $A(2, 1)$ على سطر واحد و $A(2, 4)$ و $A(2, 7)$ و $A(3, 1)$ و $A(3, 3)$ على السطر التالي ؛ $A(3, 7)$ و $A(4, 1)$ و $A(4, 4)$ و $A(4, 7)$ و على السطر الثالث ؛ $A(5, 1)$ و $A(5, 4)$ و $A(5, 7)$ على السطر الرابع .

الفصل السابع

الدوال والبرامج الصغيرة الفرعية

٧ - ١ مقدمة

قد تظهر عدة مشاكل كلما ازدادت البرامج تعقيداً أو أصبحت أكثر شمولاً فمثلاً:

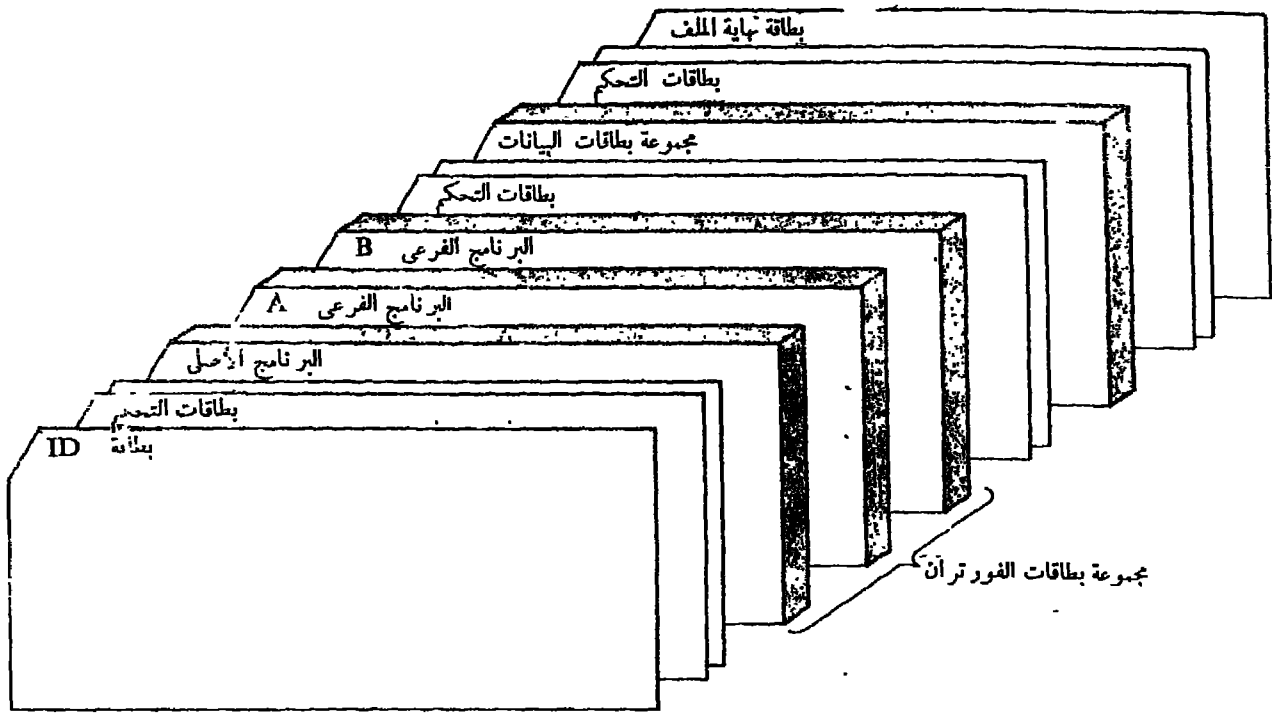
- ١ - يصعب تصميم النظام الحسابي (الخوارزم) لحل المسائل المعقدة .
- ٢ - حتى إذا كان العنصر الحسابي (الخوارزم) معروفاً فإن تنفيذه في الفورتران يصبح أكثر صعوبة وذلك لطول البرنامج .
- ٣ - كلما أصبحت البرامج أطول وأعدت ، يصبح اكتشاف الأخطاء أكثر صعوبة . وتكون البرامج الأطول أكثر تعرضاً للخطأ الذي يصعب تجنبه .
- ٤ - نحتاج إلى المزيد من توثيق البرامج حتى تصبح أسهل فهماً لأولئك الذين سيقروا ويستخدمون البرنامج .
- ٥ - قد نحتاج إلى مهام مماثلة في أجزاء متعددة من البرنامج . إذا نفذت مفردة أي واحدة تلو الأخرى فإنها تجعل المسألة المعقدة أكثر إرباكاً .

يمكن تخفيف هذه المشكلة باستخدام البرامج الفرعية . فالبرنامج الفرعي هو برنامج كامل ومستقل يمكن استخدامه (أو استدعاؤه) بواسطة البرنامج الأساسي أو بواسطة برامج فرعية أخرى . ويستقبل البرنامج الفرعي من البرنامج الأصلي (أو الداعي) قيم (تسمى خلاصات) وذلك لتنفيذ الحسابات ، وبعد ذلك تمديد (RETURNS) النتيجة أو النتائج إلى البرنامج الأصلي (أو الداعي) .

في الفصل الثاني ، ناقشنا بعض الدوال المكتوبة مثل ABS و SQRT و EXP و ALOG إلخ . إن هذه الدوال في الحقيقة هي برامج فرعية كتبت خصيصاً لحل هذه البرامج وأدجت مع بعضها في نص الفورتران نظراً لكثرة استخدامها في مواقف عديدة . وتسهيلاً لمهمة كل مستخدم لهذه البرامج الفرعية وحتى لا يضطر لكتابتها فإن مترجمات الفورتران تمدد هذه الدوال المكتوبة داخلياً . ومن الواضح ، أنه لا يمكن أن تحتوي مكتبة الفورتران على كل الدوال أو البرامج الفرعية المستخدمة في كل مجالات التطبيق ، وعلى المستفيدين أن يكتبوا برامجهم الفرعية الخاصة كلما كانت هناك حاجة لذلك . ولقد خصص هذا الفصل لمناقشة إعداد واستخدام مثل هذه البرامج الفرعية .

نذكر ترتيب حزمة الفورتران الموضحة في الشكل ١ - ٣ في قسم ١ - ٣ . يجب أن تتكون مجموعة الفورتران من البرنامج الأساسي ، بصحبة كل البرامج الفرعية . وعادة توضع البرنامج الأساسي أولاً بعد ذلك البرامج الفرعية . والشكل ٧ - ١ يوضح مجموعة الفورتران بها برنامجان فرعيان .

تقع البرامج الفرعية في مرتبتين أساسيتين : البرامج الفرعية FUNCTION والبرامج الفرعية SUBROUTINE وسيوضح في الأقسام التالية أوجه الشبه والاختلاف بين هذين النوعين من البرامج الفرعية . مع أن كلا النوعين من البرامج الفرعية يمكن ان يستقبل قيمة أو أكثر من البرنامج الأصلي (أو الداعي) فالبرنامج الفرعي FUNCTION يعود بنتيجة واحدة فقط ، في حين أن SUBROUTINE عادة يعود بأكثر من نتيجة واحدة .



شكل ٧ - ١

٧ - ٢ الدوال كبرامج فرعية

نلاحظ أن كلا من الدوال المكتوبة ABS و SQRT و SIN و ALOG ، إلخ تحسب قيمة كلما أعطيت الخلاصة المضبوطة .
افترض ، على سبيل المثال ، إذا ظهرت

$$\text{SQRT}(A^{**}2 + B^{**}2)$$

فعندما ينفذ التعبير الحسابي ، فإن الحاسب يقوم بالتالي :

- ١- ينادى على البرنامج الفرعي SQRT
 - ٢- يحسب القيمة الحقيقية $A^2 + B^2$ ويستخدم هذه القيمة كخلاصة في البرنامج الفرعي .
 - ٣- يحسب البرنامج الفرعي الجذر التربيعي لخلاصة المعطاة وهي $(A^2 + B^2)$ ويخصص هذه القيمة الحقيقية للمتغير SQRT في البرنامج الفرعي
 - ٤- يعيد قيمة SQRT من البرنامج الفرعي ثانية إلى البرنامج الأصيل (أو الداعي) ، وتموض هذه القيمة $\text{SQRT}(A^{**}2 + B^{**}2)$
- لاحظ أن هذا البرنامج الفرعي له قيمة مرتبطة باسمه SQRT . البرامج الفرعية من هذا النوع تسمى الدوال كبرامج فرعية

FUNCTION

نعطى الآن مثالين للبرامج الفرعية FUNCTION سنشير إليهما بعد ذلك عندما نذكر قواعد كتابة البرامج الفرعية .

البرنامج الفرعي الأول

ادرس مسألة إيجاد أكبر رقم من ثلاثة أرقام A و B و C . افترض أن A و B و C مخزنة في الذاكرة و BIG متغير حقيقي . سوف يستخدم لتخزين القيمة الكبرى ، شكل ٧ - ٢ (١) يبين جزء الفورتران الذي ينتهي بعد حساب BIG

```

FUNCTION BIG(A, B, C)
BIG = A
IF(BIG.LT.B) BIG = B
IF(BIG.LT.C) BIG = C
RETURN
END

```

(ب)

```

BIG = A
IF(BIG.LT.B) BIG = B
IF(BIG.LT.C) BIG = C
STOP
END

```

(أ)

شكل ٧ - ٢

افترض أننا نريد أن نكتب هذا الجزء من الفورتران كبرنامج فرعي (دالة) FUNCTION يجب أن تكون أول جملة هي جملة تعريف FUNCTION التي تخبر المترجم أن هذا هو برنامج فرعي FUNCTION. تعطى جملة تعريف FUNCTION اسم البرنامج الفرعي وخصائصه. وسنطلق على برنامجنا الفرعي BIG وله ثلاثة معاملات A و B و C وذلك يكون أول جملة في البرنامج الفرعي.

FUNCTION BIG(A, B, C)

نعتبر أن هذه المعاملات في البرنامج الفرعي قد تم تعريفها. وعلاوة على ذلك، وحيث أن هذا البرنامج الفرعي FUNCTION يستخدم بواسطة أي برنامج داعي، فإن التنفيذ لا يتوقف بجملة STOP بعد الحسابات، ولكن سيحتوي على جملة RETURN التي تنتقل التحكم مرة ثانية إلى البرنامج الداعي. يعرض شكل ٧ - ٢ (ب) جزء البرنامج مكتوب كبرنامج فرعي FUNCTION. كامل. لاحظ التشابه بين البرنامج الفرعي وجزء الفورتران شكل ٧ - ٢ (أ).

البرنامج الفرعي الثاني

الدالة $k!$ (مضروب k) تعرف بالتالي :

$$k! = \begin{cases} 1 \cdot 2 \cdot 3 \cdot \dots \cdot (k-1) \cdot k, & \text{if } k \geq 1 \\ 1 & \text{if } k = 0 \end{cases}$$

على سبيل المثال $0! = 1$ ، $4! = 1 \times 2 \times 3 \times 4 = 24$ ، $6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720$ ،

إدرس مسألة حساب $k!$ لأي رقم صحيح موجب k باعتبار k مخزنة في الذاكرة و IFACT هي متغير صحيح يستخدم لتخزين $k!$. ويعطى شكل ٧ - ٢ (أ) جزء الفورتران الذي ينتهي بعد انتهاء حساب IFACT. من ناحية أخرى، يعرض شكل ٧ - ٢ (ب) كيف نحصل على IFACT من برنامج فرعي FUNCTION لاحظ أن هناك جملة RETURN تناظر جملة STOP

```

FUNCTION IFACT(K)
C
C COMPUTES K FACTORIAL
C
IFACT = 1
IF(K.EQ.0) RETURN
DO 10 J = 1, K
    IFACT = IFACT*J
10 CONTINUE
RETURN
END

```

(ب)

```

IFACT = 1
IF(K.EQ.0) STOP
DO 10 J = 1, K
    IFACT = IFACT*J
10 CONTINUE
STOP
END

```

(أ)

شكل ٧ - ٣

نلخص الآن النقاط الرئيسية في كتابة برامج دالة فرعية FUNCTION

١- يجب أن تكون أول جملة من البرنامج الفرعي جملة تعريف FUNCTION ويكون لها الشكل التالي :

FUNCTION NAME(param1, param2, ..., paramn)

أى أن كلمة FUNCTION يتبها اسم الدالة NAME ويتبع NAME الماملات التى يفصلها عن بعضها بواسطة فاصلات وتحاط بأقواس . ويجب أن تكون هذه الماملات أسماء متغيرات بدون أدلة (أو أسماء مجموعات . زيادةً لنظر قسم ٧-٦) ، مع فرض تعريفهم مسبقاً بالنسبة للبرنامج الفرعى .

٢- تسمية البرنامج الفرعى FUNCTION تتبع نفس القواعد الخاصة بتسمية المتغيرات ؛ أى ، يجب أن تتكون من حرف واحد إلى ستة حروف أبجدية رقمية مع مراعاة أن يكون الحرف الأول أبجدياً . يجب أن يظهر اسم البرنامج الفرعى FUNCTION NAME (بدون معاملاته) أيضاً بأى طريقة فى البرنامج الفرعى (أى يظهر إما على الجانب الأيسر من جملة تخصيص أو فى جملة إدخال) قبل تنفيذ أى جملة RETURN (وبذلك يجب أن يتبع البرنامج الفرعى FUNCTION مفهوم النوع لأسماء المتغيرات) .

٣- البرنامج الفرعى هو برنامج كامل ومستقل :

(أ) كامل لأنه يحتوى على تعريفات النوع المناسبة (مثل INTEGER و REAL و DIMENSION إلخ ولأنه يمكن أن يستخدم الدوال المكتبية وبرامج فرعية أخرى وتحتوى على جملة END

(ب) مستقل لأن كل أسماء المتغيرات وكل أرقام الجمل المستخدمة داخلياً بالنسبة للبرنامج الفرعى (ماعدا الماملات) غير معروفة خارج نطاقه . وبذلك ، يمكن أن تستخدم نفس أسماء المتغيرات ونفس أرقام الجمل فى البرامج الفرعية المختلفة والبرامج التى تستدعيها .

٤- وسيلة الاتصال الوحيدة بين البرنامج الداعى والبرنامج الفرعى هى من خلال الماملات (انظر قسم ٧-٣) .

يجب أن يحتوى كل برنامج فرعى FUNCTION على معامل واحد على الأقل وجملة RETURN واحدة وهى " تنقل التحكم ثانية إلى البرنامج الداعى .

٧-٣ استدعاء البرامج الفرعية FUNCTION

يستحضر (أو يستدعى) البرنامج الفرعى FUNCTION بنفس طريقة أى دالة مكتبية . بالتحديد فى تلك الأماكن التى يراد فيها قيمة الدالة نكتب :

NAME(arg1, arg2, ..., argn)

أى ، نكتب اسم الدالة NAME متبوعاً بتعبيرات مناسبة لخلاصاتها التى تفصل عن بعضها بواسطة فاصلات وتحاط بأقواس .

ليس لزاماً أن تكون أسماء الخلاصات فى جملة الاستدعاء هى نفسها الموجودة فى الماملات المناظرة فى جملة تعريف FUNCTION . فى الحقيقة ، يمكن أن تكون الخلاصات ثوابت أو متغيرات مفردة أو تعبيرات حسابية أو حتى أسماء دوال مكتبية أو برامج فرعية أخرى كما فى الدوال المكتبية . أهم شئ يجب أن نتذكره هو أن الخلاصات يجب أن تتفق بطريقة واحد مقابل واحد فى الترتيب ونوع البيانات للماملات المناظرة فى جملة تعريف FUNCTION على سبيل المثال ، ادرس البرنامج الفرعى FUNCTION المسى BIG المذكور سابقاً وجملة تعريفه هى :

FUNCTION BIG(A, B, C)

لاحظ أن المطلوب ثلاث خلاصات حقيقية من أجل استدعاء BIG بحيث تكون الجملة التالية كلها صحيحة لئولاً .

$$\begin{aligned} \text{GRADE} &= \text{BIG}(T1, T2, T3) \\ \text{GRADE} &= \text{FINAL}/3.0 + (2.0/3.0)*\text{BIG}(T1, 60.0, \text{SQRT}(A)) \\ \text{GRADE} &= 0.5*(\text{FINAL} + \text{BIG}(S(I, 1), S(I, 2), Z(I))) \end{aligned}$$

وفما يلي نوضح المناظرة واحد مقابل واحد بين خلاصات الجملة الثانية ومعاملات البرنامج الفرعي :

T1,	60.0,	SQRT(A)	قائمة بالخلاصات
↓	↓	↓	
A,	B,	C'	قائمة بالمعاملات

هناك نقطة واحدة خفية يجب أن نفهمها عند استخدام البرامج الفرعية . وهي أن المعاملات المستخدمة في جملة تعريف الدالة FUNCTION تسمى المتغيرات الزائفة وذلك لأنه في الحقيقة لا يوجد لها أماكن تخزين . وبالتحديد ، فهي تستخدم عناوين الخلاصات المقابلة في البرنامج الداعي عند تنفيذ البرنامج الفرعي . على سبيل المثال ، في جملة الاستدعاء .

$$\text{GRADE} = \text{BIG}(T1 + T2, C, 60.0)$$

سنستخدم عناوين الخلاصات $T1 + T2$ و C و 60.0 بدلا من المعاملات A و B و C عند تنفيذ البرنامج الفرعي BIG .
ملاحظة : لاحظ أننا نستخدم المصطلح معاملات للمتغيرات في البرنامج الفرعي التي تظهر في جملة التعريف والمصطلح خلاصات القيم المناظرة لها في البرنامج الداعي .

مثال ٧ - ١

(١) يعرف المعامل ذو الحدين $\binom{n}{i}$ للأرقام الصحيحة الموجبة n و i (حيث $i \leq n$) بما يلي :

$$\binom{n}{i} = \frac{n!}{(n-i)!i!} \text{ for example, } \binom{8}{3} = \frac{8!}{5!3!} = \frac{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 1 \cdot 2 \cdot 3} = \frac{6 \cdot 7 \cdot 8}{1 \cdot 2 \cdot 3} = 56$$

اكتب البرنامج الذي يقبل أرقاما موجبة N و I (حيث $I \leq N$) يطبع القيمة $\binom{N}{I}$

باستخدام البرنامج الفرعي IFACT في القسم السابق فإن برنامج الفورتران التالي يعطينا المعامل ذا الحدين المطلوب

```

C   PROGRAM CALCULATING BINOMIAL COEFFICIENT
      INTEGER UP, DOWN
      READ(5, 10) N, I
10  FORMAT(2I8)
      UP = IFACT(N)
      DOWN = IFACT(N - I)*IFACT(I)
      IBINQ = UP/DOWN
      WRITE(6, 20) N, I, IBINQ
20  FORMAT(6X, 'N = ', I5, 5X, 'I = ', I5, 5X,
1     'BINOMIAL COEFF = ', I8)
      STOP
      END
    
```


الفصل السابع : الدول والبرامج الصغيرة الفرعية

لاحظ أن استخدام البرنامج الفرعي يجعل البرنامج الأساسي سهلاً وواضحاً . من ناحية أخرى ، إذا حسبت $n!$ و $(n-i)!$ و $i!$ مفصلة في البرنامج الأساسي ، سيكون البرنامج طويلاً ، وأقل وضوحاً وأكثر تعرضاً للخطأ . لاحظ أيضاً أن IBINQ يمكن أن تحسب بحملة واحدة :

$$IBINQ = IFACT(N)/(IFACT(N - I)*IFACT(I))$$

ومع ذلك ، فالتعبيرات الأطول والأكثر تعقيداً تكون أكثر تعرضاً للخطأ ، ومن الحكمة دائماً أن نجزي التعبيرات المعقدة إلى عدة تعبيرات بسيطة .

ملاحظة : أملنا في الواقع احتمال حدوث فيض مع $n!$ عندما تكون قيم n كبيرة ، وسنناقش فيما بعد طرق أخرى لحساب $\binom{n}{i}$.

(ب) افترض أنه تم عقد ثلاثة امتحانات قصيرة وآخر نهائى لفصل به 25 طالباً . ثقبت درجات هذه الامتحانات على صواب بجانب أرقام ID للطلبة بحيث يكون لكل طالب بطاقة واحدة . وتحسب درجة الطالب في المنهج بحساب متوسط درجة الامتحان النهائى بإضافة أعلى درجة حصل عليها في أحد الامتحانات الثلاثة القصيرة .

يحسب البرنامج التالى والذي يستخدم البرنامج الفرعى BIG درجة كل طالب .

```

WRITE(6, 10)
10 FORMAT('1', 4X, 'ID', 6X, 'SCORE 1', 2X, 'SCORE 2',
1      2X, 'SCORE 3', 6X, 'FINAL', 5X, 'GRADE')
DO 500 I = 1, 25
      READ(5, 20) ID, T1, T2, T3, FINAL
      GRADE = 0.5*(FINAL + BIG(T1, T2, T3))
      WRITE(6, 30) ID, T1, T2, T3, FINAL, GRADE
500 CONTINUE
20 FORMAT(I10, 4(F6.2, 4X))
30 FORMAT(1X, I6, 5(3X, F6.2))
STOP
END

```

ملاحظة : يمكن أن يستدعى أى برنامج فرعى برنامجاً فرعياً آخر ، ولكن لا يمكن أن يستدعى نفسه . وبصورة أعم لا يمكن أن تشكل البرامج الداعية دائرة . فمثل سبيل المثال لا يمكن أن تعرض الحالة التالية :

البرنامج الفرعى A يستدعى البرنامج الفرعى B

البرنامج الفرعى B يستدعى البرنامج الفرعى C

البرنامج الفرعى C يستدعى البرنامج الفرعى D

البرنامج الفرعى D يستدعى البرنامج الفرعى A

وحتى نفسين، عدم حدوث ذلك يجب أن نضع البرنامج الفرعى قبل أى برنامج آخر يستدعيه ، مثلاً إذا كان البرنامجان الفرعيان X و Y يستدعيان البرنامج الفرعى Z فنضع Z بعد X و Y في مجموعة بطاقات برامج فورتران .

٧ - ٤ دوال البرامج الفرعية FUNCTION الحاسبة لعدة قيم

إلى الآن ، كل برنامج الفرعية FUNCTION كانت تحسب قيمة واحدة ، وكانت تخصص هذه القيمة لإسم الدالة NAME في البرنامج الفرعي . سؤال : هل يستطيع برنامج فرعي FUNCTION أن يحسب عدة قيم ؟ الرد هو « نعم » (رغم أن البرنامج الفرعي SUBROUTINE والذي سيناقش في قسم ٧ - ٧ هو الذي يستخدم عادة في مثل هذه الحالة) .

تذكر أن وسيلة الاتصال الوحيدة بين البرنامج الداعي والبرنامج الفرعي تكون من خلال الخلاصات والمعاملات . وبالتالي ، كلما استخدمنا برنامجاً فرعياً FUNCTION لحساب عدة قيم ، تخصص قيمة واحدة لإسم الدالة NAME والقيم الأخرى يجب أن تخصص إلى معاملات زائفة (وبالتالي تنقل أيضاً إلى البرنامج الداعي) .

مثال ٧ - ٢

افرض أننا نريد أن نكتب البرنامج الفرعي FUNCTION الذي يحسب أكبر رقم ومجموع أي ثلاثة أرقام A و B و C . يسمى البرنامج BBB حيث تستخدم BBB لتخزين القيمة الكبرى . سيكون البرنامج الفرعي مشابهاً للبرنامج الفرعي BIG في شكل ٧ - ٢ فيما عدا أنه يجب أن يكون لـ BBB معامل رابع وليكن SUM الذي سوف يستخدم لنقل مجموع A و B و C إلى البرنامج الداعي . ويظهر مثل هذا البرنامج BBB كالتالي :

```
FUNCTION BBB(A, B, C, SUM)
SUM = A + B + C
BBB = A
IF(BBB.LT.B) BBB = B
IF(BBB.LT.C) BBB = C
RETURN
END
```

مع فرض أن البرنامج الداعي به الجملة التالية :

```
RESULT = BBB(X, Y, Z, TOTAL)
```

بعد تنفيذ هذه الجملة ، ستحتوي RESULT على أكبر الأرقام X و Y و Z وستحتوي TOTAL على مجموعها . ستكون TOTAL بعد ذلك متاحة للحسابات . على سبيل المثال ، إذا نفذت فيما بعد الجملة التالية :

```
AVE = TOTAL/3.0
```

فسوف تحتوي AVE على متوسط X و Y و Z

٧ - ٥ مجموعات متراسة وبرامج فرعية FUNCTION ، أبعاد متغيرة

يمكن أيضاً أن تستخدم مجموعة متراسة كعامل لبرنامج فرعي FUNCTION . في مثل هذه الحالة ، يجب أن تكون الخلاصة المناظرة في جملة الاستدعاء مجموعة متراسة أيضاً . ومع ذلك يجب تعريف المجموعة المتراسة في البرنامج الفرعي بجملة DIMENSION وذلك في البرنامج الفرعي . وعلاوة على ذلك فلا يمكن أن تتجاوز أبعاد الخلاصة المناظرة .

شكل ٧ - ٤ (أ) عبارة عن برنامج فرعي BIGG يحدد أكبر عنصر في المجموعة المتراسة الخطية A التي بها 25 عنصراً ، وشكل ٧ - ٤ (ب) عبارة عن برنامج استدعاء على .

سيحدد هذا البرنامج الفرعي BIGG العنصر الأكبر في مجموعة متراسة خطية بها بالتحديد 25 عنصراً ، وهذا من البديهي يحدد من استخدامها . يسمح الفورتران بمرونة أكبر وذلك باستخدام الأبعاد المتغيرة (أي ، الأبعاد القابلة للتعديل) وذلك في البرامج الفرعية فقط . وسناقش الآن هذه الخاصية .

```

DIMENSION X(25)
REAL LARGE
READ(5, 10) X
10 FORMAT(5(F8.2, 2X))
LARGE = BIGG(X)
WRITE(6, 20) LARGE
20 FORMAT('0', 'LARGEST VALUE IS', 2X, F8.2)
STOP
END

```

(ب)

```

FUNCTION BIGG(A)
DIMENSION A(25)
BIGG = A(1)
DO 10 K = 2, 25
IF(BIGG.LT.A(K)) BIGG = A(K)
10 CONTINUE
RETURN
END

```

(أ)

شكل ٧ - ٤

الأبعاد المتغيرة . نتذكر أنه في أي برنامج يجب أن يكون دليل اسم المجموعة المتراسة في جملة DIMENSION ثابتاً صحيحاً ولا يمكن أن يكون متغيراً . ويكون هذا غير صحيح تماماً إذا ظهرت جملة DIMENSION في برنامج فرعي . بالتحديد افترض أن جملة DIMENSION في برنامج فرعي : أن المتغير A مجموعة متراسة و A هو معامل البرنامج الفرعي . إذن يمكن أن يكون دليل A في جملة DIMENSION متغيراً صحيحاً بشرط أن يكون المتغير أيضاً معاملاً في البرنامج الفرعي . ويتم توضيح ذلك في البرنامج الفرعي FUNCTION التالي الذي يحدد أكبر عنصر في متجه A عدد عناصره N .

```

FUNCTION BIGMM(A, N)
DIMENSION A(N)
BIGMM = A(1)
DO 10 K = 1, N
IF(BIGMM.LT.A(K)) BIGMM = A(K)
10 CONTINUE
RETURN
END

```

ويجدر أن نؤكد أن استخدام الأبعاد المتغيرة مسموح به فقط في البرامج الفرعية ، أي ، لا يمكن استخدامها بأي حال من الأحوال في البرنامج الأساسي .

ملاحظة : لا يجب أن تتجاوز قيمة المعامل N المستخدم في جملة DIMENSION المتغيرة في برنامج فرعي حجم البعد الأصل للمجموعة المتراسة المناظر في البرنامج الداعي .

مثال ٧ - ٣

(أ) اكتب البرنامج الفرعي FUNCTION الذي يحسب مجموع العناصر في مجموعة متراسة خطية بها N عناصر . هنا A و N معاملات :

```

FUNCTION SUM(A, N)
DIMENSION A(N)
SUM = 0.0
DO 99 K = 1, N
SUM = SUM + A(K)
99 CONTINUE
RETURN
END

```

(ب) افرض أن X و Y مجموعات مترابطة خطية مخزنة في الذاكرة ومعرفة كالآتي

DIMENSION X(100), Y(200)

استخدام التعريف السابق للدالة SUM لإيجاد المتوسطات التالية :

$$(X_1 + X_2 + \dots + X_{25})/25$$

$$(Y_1 + Y_2 + \dots + Y_M)/M$$

فيصبح لدينا :

$$AVEX = \text{SUM}(X, 25)/25.0$$

$$AVEY = \text{SUM}(Y, M)/\text{FLOAT}(M)$$

(افترضنا أن $M \leq 200$)

٧ - ٦ دوال الجملة الحسابية

افرض أننا نريد أن نحسب قيمة الدالة التربيعية التالية :

$$g(x) = x^2 - 5x + 2$$

لقيم X حيث : $x = 1, 2, \dots, 20$. باستخدام برنامج FUNCTION يصبح لدينا :

<pre> DO 100 J = 1, 20 VALUE = G(FLOAT(J)) WRITE(6, 10) J, VALUE 100 CONTINUE 10 FORMAT(1X, I3, 3X, F8.2) STOP END </pre>	<pre> FUNCTION G(X) G = X*X - 5.0*X + 2.0 RETURN END </pre>
--	---

لاحظ أن هذا البرنامج الفرعي FUNCTION يتكون من جملة حسابية واحدة (إلى جانب جملة التعريف FUNCTION وجملتنا RETURN و END) يمكن أن نبسط مثل هذا النوع من البرامج الفرعية FUNCTION ببساطة بكتابة الجملة

$$G(X) = X*X - 5.0*X + 2.0$$

في بداية البرنامج الداعي كما يلي :

```

C
C   DEFINE THE FUNCTION G
C
C   G(X) = X*X - 5.0*X + 2.0
C
C   COMPUTE FUNCTIONAL VALUES
C
C
DO 100 J = 1, 20
  VALUE = G(FLOAT(J))
  WRITE(6, 10) J, VALUE
100 CONTINUE
10  FORMAT(1X, I3, 3X, F8.2)
STOP
END

```

وتسمى الدالة G دالة الجملة الحسابية .

وفيما يلي قواعد دالة الجملة الحسابية :

١ - تعرف دالة الجملة باستخدام الشكل التالي :

تعبير حسابي = اسم (معامل 1 ، ومعامل 2 ، ... ، ومعامل n)

يجب أن تكون المعاملات متغيرات بدون أدلة ، ولا يجب أن يحتوى التعبير الحسابي على أى متغيرات بدليل . ومع ذلك ، يمكن أن يحتوى التعبير الحسابي على دوال مكتوبة وعلى برامج فرعية FUNCTION وعلى دوال جمل أخرى (بشرط أن تكون قد سبق تعريفها) . رغم أن بعض المترجمات يسمح بظهور جملة التعريف في أى مكان في البرنامج بشرط أن يكون سابقاً لاستخدامها لكن بعض المترجمات الأخرى تتطلب أن تكون موضوعة في بداية البرنامج بعد أى جمل نوع أو تعريف ، ولكن قبل أى جملة قابلة للتنفيذ .

٢ - تتبع تسمية الدالة ومعاملاتها نفس قواعد البرامج الفرعية FUNCTION والمعاملات هي ، كما في البرامج الفرعية FUNCTION متغيرات زائفة ، ولا يتم تخصيص ما تثن تخزين لها في الذاكرة وبذلك تكون خاصة بالجملة .

٣ - قواعد استدعاء دوال الجمل هي نفس القواعد المستخدمة للبرامج الفرعية FUNCTION . أى في أى مكان من البرنامج يتطلب فيه قيمة الدالة يكتب ببساطة اسم دالة الجملة بملصقات مناسبة . ويمكن أن تكون الملصقات أى تعبيرات حسابية طالما أنها متفقة بطريقة واحد إلى واحد في الترتيب ونوع المعاملات مع جملة التعريف . على سبيل المثال ، باعتبار دالة الجملة السابقة G(X) قد عرفت في البرنامج ، يمكن أن أى تستدعى كما يلي :

$$AVE = (G(A + SQRT(B)) + DEP)/2.0$$

حيث AVE هي متوسط DEP و G تحسب قيمتها عند A + SQRT(B) . الخلاصة هنا هي (A + SQRT(B)) وهي تقابل المعامل X في جملة الدالة .

٤ - نقطة هامة هي أن دالة الجملة داخلية في البرنامج (أو البرنامج الفرعي) التي تظهر فيه ، من ثم ، لا يمكن أن تستدعى بأى برنامج آخر .

حيث أن دالة الجملة داخلية لبرنامجها فلها درجة حرية ليست لدى البرامج الفرعية FUNCTION . بالتحديد . المتغيرات التي لا تظهر في قائمة المعاملات يمكن أن تظهر في الجملة . على سبيل المثال ، دالة الجملة الحسابية الآتية صحيحة لغوياً :

$$F(X, Y) = A*X**2 + B*X*Y + C*Y**2$$

في هذه الحالة X و Y متغيرات زائفة ، ومن ثم ، يمكن أن تستخدم X و Y كأسماء متغيرات في مكان آخر في البرنامج . ومع ذلك فإن A و B و C ليست متغيرات زائفة حيث أنها ليست في قائمة المعاملات ، ولذلك سوف نفترض أنه تم تعريفها عند نداء الدالة . على سبيل المثال :

$$F(1.0, 2.0)$$

ستحسب الدالة لقيم A و B و C الحالية .

مثال ٧ - ٤

نفرض مجموعة N من البطاقات ، كل بطاقة مثقب عليها ثلاثة أرقام حقيقية ، هي المعاملات A وB وC لمعادلة من الدرجة الثانية :

$$F(X) = AX^2 + BX + C$$

اكتب برنامج فورتران لحساب قيمة الدالة عندما $X = -5, -4, \dots, 5$ لكل ثلاثى A وB وC .

لاحظ أن هناك 11 قيمة لـ X ونستخدم حلقة DO بدليل $I = 1, 2, \dots, 11$ والملاقة بين المعامل I وقيم X يمكن أن تعرض كالتالي :

I: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
X: -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5

لاحظ أن I و X تربطهما الملاقة $I - X = 6$ من ثم :

$$X = -6 + I$$

فيما يلي البرنامج المطلوب :

```

C
C      EXAMPLE USING STATEMENT FUNCTION
C
      F(X) = A*X*X + B*X + C
      READ(5, 5) N
5     FORMAT(I3)
      DO 100 J = 1, N
          READ(5, 10) A, B, C
10    FORMAT(3F5.1)
          WRITE(6, 15) A, B, C
15    FORMAT('1', 'A = ', F5.1/1X, 'B = ', F5.1/1X, 'C = ', F5.1)
C
C      EVALUATING THE FUNCTION
C
      DO 200 I = 1, 11
          Y = F(-6.0 + FLOAT(I))
          WRITE(6, 20) I, Y
200   CONTINUE
      FORMAT(1X, 3I, 2X, F8.2)
100   CONTINUE
      STOP
      END
    
```

٧ - ٧ برامج صغيرة فرعية SUBROUTINES

هناك نوع آخر من البرامج الفرعية يسمى برنامجاً صغيراً فرعياً SUBROUTINE . نذكر ما سبق شرحه أن اسم البرنامج الفرعي FUNCTION تخصص له دائماً قيمة في البرنامج الفرعي . هذا ليس صحيحاً بالنسبة للبرامج الصغيرة الفرعية SUBROUTINE . وما عدا هذا الفرق وما يترتب عليه ، تتشابه القواعد والقيود المتحركة في كتابة البرامج الصغيرة الفرعية SUROUTINE مع تلك الخاصة بالبرامج الفرعية FUNCTION يجب أن نلاحظ أننا نستخدم SUBROUTINE

أكثر من البرنامج الفرعي FUNCTION عندما يكون لدينا قيم متعددة نريد حسابها . علاوة على حساب قيم متعددة ، نستخدم البرامج الصغيرة الفرعية SUBROUTINE لإنجاز مهام (مثل استبدال عناصر .. إلخ) .

كما هو المثل ، فإن البرامج الفرعية FUNCTION يجب الإعلان أولاً عن البرامج الصغيرة الفرعية SUBROUTINE باستخدام جملة التعريف SUBROUTINE ويأخذ الإعلان عن البرنامج الفرعي الصورة العامة التالية :

SUBROUTINE NAME(param1, param2, ..., paramn)

وحيث أن اسم البرنامج الفرعي الصغير (NAME) لا تخصص له قيمة في البرنامج الفرعي فلا داعي لمناقشة موضوع الاسم هنا . (ومع ذلك لا يمكن أن تتجاوز أسماء البرامج الفرعية الصغيرة ستة حروف) .

وهو كأي برنامج فرعي (Subprogram) ، فإن SUBROUTINE أيضاً برنامج كامل ومستقل . من ثم ، يمكن أن يحتوي على كل خصائص أى برنامج : إعلانات النوع وجملة DIMENSION وبرنامج فرعي FUNCTION أو وبرنامج فرعي صغيرة SUBROUTINE وجملة END وهكذا . وعلاوة على ذلك فالاتصال بين البرنامج الداعي والبرنامج الصغير الفرعي SUBROUTINE يكون فقط من خلال المعاملات ، ومن ثم فأسماء المتغيرات (فضلاً عن المعاملات) و أرقام الجمل في SUBROUTINE تعتبر عملية (داخلية) لهذا البرنامج الفرعي . وسنناقش في جزء متأخر من هذا القسم طريقة استدعاء SUBROUTINE وطريقة نقل القيم من خلال المعاملات) .

نعطي الآن مثالين للبرامج الصغيرة الفرعية SUBROUTINE سيؤدي الأول مهمة وسيحسب الثاني قياً .

١- بفرض أن X و Y قد سبق تعريفهما يستبدل فجزء البرنامج التالي القيم في X و Y

```
T = X
X = Y
Y = T
```

نستطيع أن نكتب برنامجاً صغيراً فرعياً SUBROUTINE ويسمى INTCHG لتأدية عملية التبدل :

```
.SUBROUTINE INTCHG(X, Y)
T = X
X = Y
Y = T
RETURN
END
```

لاحظ أولاً أن الاسم INTCHG لا تخصص له قيمة في البرنامج الصغير الفرعي (وعلى ذلك فبده الاسم بالحرف I لا يهم) يفرض أن القيم المستبدلة سوف تستخدم قياً بعد في البرنامج الداعي لدينا جملة RETURN التي تنقل التحكم ثانية إلى البرنامج الداعي .

٢- ادرس مرة أخرى مسألة كتابة البرامج الفرعية الذي يحسب القيمة الكبرى ومجموع ثلاثة أرقام A و B و C يمكن عمل هذا بالبرنامج الصغير الفرعي SUBROUTINE حيث نستخدم الخلاصات لنقل القيمة المحسوبة ثانية إلى البرنامج الداعي . فيما يلي مثال لهذا البرنامج الفرعي SUBROUTINE :

```
SUBROUTINE LARGE(A, B, C, BIG, SUM)
SUM = A + B + C
BIG = A
IF(BIG.LT.B) BIG = B
IF(BIG.LT.C) BIG = C
RETURN
END
```

لاحظ التشابه والفرق بين البرنامج الفرعي هذا وبين البرنامج الفرعي FUNCTION في مثال ٧-٢ . الاسم LARGE هنا ليس له قيمة ، ومن ثم ، BIG الأكبر بين A و B و C يجب أن تذكر كعامل حتى يمكن نقلها ثانية إلى البرنامج الداعي .

كيف نستدعي البرامج الصغيرة الفرعية SUBROUTINE ؟ كما سبق أن ذكرنا هناك اختلافاً عن البرنامج الفرعي FUNCTION فاسم البرنامج الفرعي SUBROUTINE ليست له قيمة ، وعلى ذلك فطلب نوع جديدمن الجمل - جملة CALL فجملة النداء التالية تستدعي البرنامج الصغير الفرعي INTCHG

CALL INTCHG(A(3), T)

والتأثير النهائي لهذه الجملة في البرنامج الداعي هو استبدال القيم A(3) و T

نلاحظ أنه بعد استدعاء INTCHG تنقل عناوين A(3) و T إلى البرنامج الفرعي ويعوض بهما فيه بدلا عن المتغيرات الزائفة X و Y . حقيقة أن الاسم T في البرنامج الداعي مستخدمة أيضاً كاسم في البرنامج الفرعي INTCHG وذلك لا يجب أى تشويش لأن المتغير T في SUBROUTINE داخل هذا البرنامج الفرعي ويستخدم في تنفيذ SUBROUTINE عنوان الخلاصة الداعية T (وليس الاسم T) .

جملة النداء التالية صحيحة لغوياً أيضاً :

CALL LARGE(U + V, W - SQRT(V), U*U, T, S)

التأثير النهائي لهذه الجملة في البرنامج الداعي هو أن S ستحتوي على المجموع U + V و W - SQRT(V) و U*U وستحتوي T على أكبر قيمة من القيم الثلاثة .

أنا نؤكد مرة ثانية ، كما في حالة البرامج الفرعية FUNCTION أنه يجب أن نتفق الخلاصات بطريقة واحد - إلى واحد - وجملة النداء مع ترتيب ونوع المعاملات في جملة التعريف SUBROUTINE نوضح هذه الحقيقة باستخدام جملة CALL السابقة :

U + V,	W - SQRT(V),	U*U,	T,	S	(أ) قائمة الخلاصات
↓	↓	↓	↓	↓	
A,	B,	C,	BIG,	SUM	(ب) قائمة المعاملات

مثال ٧-٥

افرض أن A مجموعة متراصة تحتوى على N عنصر على الأكثر . اكتب البرنامج الصغير الفرعي SUBROUTINE الذى :

١ - يحدد أكبر قيمة من عدد العناصر K الأولى :

A(1), A(2), ..., A(K).

٢ - يحسب مجموع عدد العناصر K الأولى .

A(1) + A(2) + ... + A(K)

ينجز البرنامج الصغير الفرعي SEEK ما سبق التوصل إليه :

```

SUBROUTINE SEEK(A, N, K, HIGH, SUM)
DIMENSION A(N)
SUM = 0.0
HIGH = A(1)
DO 10 J = 1, K
    SUM = SUM + A(J)
    IF(HIGH.LT.A(J)) HIGH = A(J)
10 CONTINUE
RETURN
END

```

كما نرى في المثال السابق ، يمكن استخدام أسماء مجموعات متراسة كعلامات في برنامج فرعي SUBROUTINE (بشرط أن تظهر أيضاً في جملة DIMENSION في البرنامج الفرعي) ، ويمكن أن تأخذ ابتداءً متغيرة . أى تتشابه القواعد المتحركة في المجموعات المتراسة في البرامج الفرعية SUBROUTINE مع تلك الخاصة بالبرامج الفرعية FUNCTION :

مثال ٧ - ٦

افرض أن A مجموعة متراسة بها N عنصر مخزنة في ترتيب تصاعدي ، وبالتحديد $A(1) \leq A(2) \dots \leq A(N)$. اكتب برنامجاً صغيراً فرعياً SUBROUTINE ليبيد ترتيب عناصر المجموعة المتراسة في ترتيب تنازلي ، أى $A(1) \geq A(2) \dots \geq A(N)$. يمكننا بكل تأكيد فرز المجموعة المتراسة في ترتيب تنازلي ، إلا أن هذه العملية ستكون كفاءتها منخفضة لأن A كان لم - مرتبة ترتيباً تصاعدياً . وأبسط الطرق لتنفيذ هذه المهمة هي قلب المجموعة المتراسة ، أى نستبدل A(1) و A(N) ونستبدل A(2) و A(N-1) وهكذا . ونطلق على هذا البرنامج الصغير الفرعي INVERT وهو يستدعي البرنامج الصغير الفرعي INTCHG :

```

SUBROUTINE INVERT(A, N)
DIMENSION A(N)
NN = N/2
K = N + 1
DO 20 I = 1, NN
    CALL INTCHG(A(I), A(K - I))
20 CONTINUE
RETURN
END

```

(لاحظ جملة DO هل يستطيع القارئ أن يقول السبب في أن حلقة DO تستمر إلى N/2 فقط بدلا من أن تستمر إلى N ؟)

يقطع استخدام البرامج الفرعية المشكلة الكبيرة والمعقدة إلى وحدات برامج أصغر . تكون كل وحدة صغيرة أسهل في التنفيذ واكتشاف وتصحيح الأخطاء . علاوة على ذلك ، يمكن أيضاً تنفيذ وحدات البرامج الصغيرة بواسطة عدد من أفراد الفريق . وبالتالي ، يمكن تنفيذ عملية كتابة برنامج كبير على التوازي بدلا من أن تتم على التوالي .

٧ - ٨ مقارنة SUBROUTINE مع FUNCTION

تستخدم البرامج الفرعية FUNCTION بصفة عامة ، لحساب قيمة واحدة ، في حين تستخدم البرامج الفرعية SUBROUTINE لحساب عدة قيم أو تنفيذ مهام محددة (مثل استبدال قيم) . نلاحظ أي شيء ينفذ برنامج فرعي FUNCTION يمكن أن يتم برنامج فرعي SUBROUTINE . وبالعكس أيضا معظم المهام المنفذة ببرامج فرعية SUBROUTINE يمكن تنفيذها ببرامج فرعية FUNCTION على سبيل المثال ، نجد أن البرنامج الفرعي FUNCTION لتبديل قيم X و T كما يلي :

```

FUNCTION EXCHG(X, Y)
  T = X
  X = Y
  Y = T
  EXCHG = 0.0
  RETURN
END

```

لاحظ أن اسم الـ FUNCTION يُخصص له قيمة اختيارية نظراً لأنه يجب أن يعرف في البرنامج الفرعي . وحيث أن الاتصال يتم عن طريق المعاملات ، فإن قيم X و Y مستبدل في البرنامج الأساسى . إلا أن هذه التركيبة صناعية نوعاً ما وغير طبيعية . وفيما يلي الفروق الأساسية بين البرامج الفرعية SUBROUTINE و FUNCTION :

- ١ - لا تخصص قيمة لاسم البرنامج الفرعي الصغير (NAME) SUBROUTINE ، وحيث أنه يجب أن تكون هناك قيمة عددية أو منطقية لاسم البرنامج الفرعي FUNCTION ، من ثم ، يجب أن ينطبق على اسم (NAME) البرنامج الفرعي FUNCTION مفهوم النوع ، ويجب أن يعرف في البرنامج الفرعي .
- ٢ - يمكن أن يستدعى SUBROUTINE فقط بواسطة جملة نداء خاصة وهى جملة CALL . يجب أن يستخدم اسم البرنامج الفرعي FUNCTION بنفس طريقة الدوال المكتوبة . أى ، فى تميزات حسابية ، إلخ .
- ٣ - يجب أن يكون البرنامج الفرعي FUNCTION على الأقل خلاصة واحدة ، بينما يمكن ألا تكون هناك أى خلاصات للبرنامج الفرعي الصغير SUBROUTINE (انظر مسألة ٧ - ١١) .
- ٤ - حيث أن البرنامج الفرعي FUNCTION يجب على الأقل قيمة واحدة يجب أن يحتوى على جملة RETURN أما البرنامج الفرعي الصغير SUBROUTINE فيمكن ألا يحتوى على جملة RETURN (انظر مسألة ٧ - ١٣) .

مسائل محلولة

برامج فرعية صغرية

٧ - ١ اكتشف الأخطاء ، إن وجدت ، فى كل من جمل تعريف البرامج الفرعية الصغرية الآتية :

(أ) FUNCTION, NEXT(A, B + C, X) (ب) SUBROUTINE NEW(X, Y(3), Z)

(ب) FUNCTION GRADE A, B, C (د) SUBROUTINE(U, V, W)

(أ) لا يجب أن تكون هناك فصلة بعد FUNCTION . أيضاً لا يمكن أن تكون $B + C$ معامل .

(ب) يجب أن تحاط المعاملات A و B و C بأقواس .

(ج) لا يمكن أن يكون المتغير ذو الدليل $Y(3)$ معامل فى جملة تعريف برنامج فرعى .

(د) البرنامج الفرعى الصغير ليس له اسم .

٧ - ٢ اكتشف الأخطاء ، إن وجدت ، في كل جزء من الأجزاء التالية :

(١) FUNCTION XXX(A, N, K) (ب) SUBROUTINE YYY(A, N, K)
DIMENSION A(N, N + 1), B(K) DIMENSION A(J, K), B(N)

(١) لا يمكن أن يظهر التعبير الحساب $N + 1$ في جملة DIMENSION

(ب) لا يمكن أن يظهر المتغير J في جملة DIMENSION حيث أنه لا يظهر كعامل في جملة التعريف .

٧ - ٣ اكتشف الأخطاء ، إن وجدت ، في كل برنامج فرعي مما يلي :

(١) FUNCTION ADD(X, Y, Z) (ب) SUBROUTINE AAA(X, Y, Z)
 $Z = X + Y$ $Z = X + Y$
RETURN RETURN
END END

(١) يجب تخصيص قيمة لاسم الدالة ADD في البرنامج الفرعي ، ولم يحدث هذا .

(ب) لا توجد أخطاء (لاحظ عدم تخصيص قيمة لاسم البرنامج) .

٧ - ٤ عرف دالة الجملة لحساب $R = \sqrt{u^2 + v^2 + w^2}$: واستخدمها لحساب قيمة

$$A = \frac{x}{\sqrt{x^2 + y^2 + z^2}}, \quad B = \sqrt{x^4 + y^4 + z^4}, \quad C = \sqrt{4x^2 + 9y^2 + 4z^2}$$

نلاحظ أن $x^4 = (x^2)^2$ و $4x^2 = (2x)^2$ وهكذا .

$R(U, V, W) = \text{SQRT}(U*U + V*V + W*W)$
 $A = X/R(X, Y, Z)$
 $B = R(X*X, Y*Y, Z*Z)$
 $C = R(2.0*X, 3.0*Y, 2.0*Z)$

٧ - ٥ أوجد المخرجات للبرامج التالية والتي تستخدم دالة الجملة :

(١) $\text{JF}(M) = M**2 - 3*M + 4$
 $K = 2$
 $L = \text{JF}(K + 2)$
 $M = \text{JF}(L - 3*K) + K$
WRITE(6, 10) K, L, M
10 FORMAT(1X, 3(I10, 2X))
STOP
END

(ب) $\text{F}(X, Y) = A*X + B*Y$
 $X = 2.0$
 $Y = 3.0$
 $A = 4.0$
 $B = 5.0$
 $C = \text{F}(6.0, 7.0)$
WRITE(6, 10) C
10 FORMAT(1X, F10.2)
STOP
END

(١) يعرف السطر الأول للدالة $\text{JF}(M) = M^2 - 3M + 4$ ويخصص السطر الثاني 2 إلى K . وتحسب قيمة L و M بواسطة السطرين الثالث والرابع كما يلي :

$$L \leftarrow \text{JF}(2 + 2) = \text{JF}(4) = 4^2 - 3 \cdot 4 + 4 = 8$$

$$M \leftarrow \text{JF}(8 - 3 \cdot 2) + 2 = \text{JF}(2) + 2 = (2^2 - 3 \cdot 2 + 4) + 2 = 4$$

ومن تم ستطبع 2 و 8 و 4 على سطر واحد ، كل في حقل بعرض 10 .

(ب) تخصص 2 و 3 و 4 و 5 إلى X و Y و A و B على الترتيب وذلك بواسطة الأسطر من الثاني إلى الخامس . تذكر أن X و Y في $F(X, Y)$ متغيرات زائفة وليس لها علاقة بالقيم X و Y في البرنامج . بالتحديد ، عند حساب قيمة C بواسطة السطر السادس وباستخدام قيم A و B في البرنامج ، نعوض عن X بالرقم 6 وعن Y بالرقم 7 في $F(X, Y)$ وبذلك يصبح :

$$C \leftarrow 4 \cdot 6 + 5 \cdot 7 = 59$$

وبالتالي ، ستطبع 59.00 مضبوطة من الطرف الأيمن في أول 10 أعمدة من صفحة الطباعة .

٧-٦ افترض أننا أعطينا البرامج الفرعية الآتية :

FUNCTION SUM(X, Y, Z)	SUBROUTINE ADD(X, Y, Z, TOTAL)
SUM = X + Y + Z	Y = X + Y
Y = X + Y	Z = Y + Z
Z = Y + Z	X = Z + X
X = Z + X	TOTAL = X + Y + Z
RETURN	RETURN
END	END

أوجد المخرج لجزء البرنامج التالي :

```

A = 1.0
B = 2.0
C = 3.0
AMOUNT = SUM(A, B, C)
CALL ADD(A, B, C, GT)
WRITE(6, 10) A, B, C, AMOUNT, GT
10 FORMAT(1X, 5(F4.1, 2X))

```

بعد تنفيذ السطر الرابع يصبح لدينا $A = 7.0$ و $B = 3.0$ و $C = 6.0$ و $AMOUNT = 6.0$. وبعد تنفيذ السطر الخامس يصبح لدينا $A = 23.0$ و $B = 10.0$ و $C = 16.0$ و $GT = 49.9$ ومن ثم يتكون المخرج من 23.0 و 10.0 و 16.0 و 6.0 و 49.0

برامج

٧-٧ اكتب البرنامج الفرعي SUBROUTINE الذي يحسب الرصيد الجديد لحساب مراجعة ، ليكن في الشكل :

```
BANK(BAL, DEP, M, CK, N)
```

حيث :

BAL تدل على الرصيد الشهري لحساب المراجعة

و DEP مجموعة مزاولة تعطى قائمة M بودائع الشهر .

و CK مجموعة مزاولة تعطى قائمة بالمدفوعات على عدد N من الشيكات خلال الشهر .

افترض أن هناك مقابل خدمة (SC) عبارة عن \$1.00 شهرياً و 5¢ لكل شيك و 2¢ لكل وديعة . أيضاً اعتبر أن DEP و CK لا تتجاوز أبداً 100 عنصر .

نجمع ببساطة الروائع ونطرح الشيكات ومقابل الخدمة . ومع ذلك يجب أيضاً أن نأخذ في الاعتبار الحالة التي تكون فيها M أو N أصفار . وفيما يلي البرنامج الفرعي :

```

SUBROUTINE BANK(BAL, DEP, M, CK, N)
DIMENSION DEP(100), CK(100)
IF(M.EQ.0) GO TO 10
DO 100 K = 1, M
    BAL = BAL + DEP(K)
100 CONTINUE
10 IF(N.EQ.0) GO TO 20
DO 200 K = 1, N
    BAL = BAL - CK(K)
200 CONTINUE
20 SC = 1.00 + M*0.02 + N*0.05
    BAL = BAL - SC
RETURN
END

```

٧-٨ اكتب البرنامج الفرعي SUBROUTINE الذي ليس له أي (١) خلاصة (ب) جملة RETURN

(١) فيما يلي برنامج فرعي SUBROUTINE بدون أي خلاصة .

```

SUBROUTINE NEW
WRITE(6, 10)
10 FORMAT(1X, 'NEW DEPOSITOR')
RETURN
END

```

وتكون جملة الاستدعاء من البرنامج كما يلي :

```
CALL NEW
```

ويكون لها تأثير طباعة الرسالة التالية :

```
NEW DEPOSITOR
```

(ب) لا يحتاج البرنامج الفرعي SUBROUTINE أن يرجع إلى البرنامج الداعي . بالتحديد ، لو وضعنا في مكان جملة RETURN في أي برنامج فرعي SUBROUTINE جملة STOP سينتهي تنفيذ برنامج الفورتران مع SUBROUTINE على سبيل المثال ، لو وضعنا STOP مكان RETURN في (١) سيتوقف البرنامج بعد طباعة NEW DEPOSITOR

٧-٩ افترض أن A و B و C و D مجموعات متراصة خطية مخزنة في الذاكرة لما 100 و 50 و 75 و 200 عنصر على الترتيب . افترض L و K مخزنة أيضاً في الذاكرة $5 \leq L \leq 75$ و $L < K \leq 200$ اكتب جزء برنامج فورتران لحساب ما يلي :

1. $A_{15} + A_{17} + A_{19} + \dots + A_{77}$
2. $B_{22} + B_{23} + B_{24} + \dots + B_{36}$
3. $C_2 + C_4 + C_8 + \dots + C_L$
4. $D_L + D_{L+1} + D_{L+2} + \dots + D_K$

لاحظ أن كلا من الحسابات تشتمل على إيجاد مجموع حدود في مجموعة متراصة خطية . وبدلاً من عمل كل من الحسابات في البرنامج نكتب دالة برنامج فرعي تدعى SUM لحساب $X_{1N} + X_{1N+1C} + \dots + X_{1T}$ حيث X

مجموعة متراسة خطية بها N عنصر ، IN الدليل الابتدائي و IC معامل الزيادة و IT الدليل النهائي . وفيما يلي البرنامج الفرعي الذي ينفذ ذلك وكذا جزء برنامج الفورتران الذي يستدعيه .

<p>جزء برنامج الفورتران</p> <p>T1 = SUM(A, 100, 15, 77, 2) T2 = SUM(B, 50, 22, 36, 1) T3 = SUM(C, 75, 2, L, 3) T4 = SUM(D, 200, L, K, 1)</p>	<p>البرنامج الفرعي</p> <p>FUNCTION SUM(X, N, IN, IT, IC) DIMENSION X(N) SUM = 0.0 DO 20 I = IN, IT, IC SUM = SUM + X(I) 20 CONTINUE RETURN END</p>
---	---

٧-١٠ افرض X و Y مجموعات متراسة خطية كل بها N عنصر . اكتب جزء برنامج لحساب ما يلي

$$\frac{\sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \cdot \sqrt{y_1^2 + y_2^2 + \dots + y_n^2}}{\sqrt{x_1 y_1 + x_2 y_2 + \dots + x_n y_n}}$$

باستخدام (FUNCTION) برنامج فرعي INNPRO الذي يجب $a_1 b_1 + a_2 b_2 + \dots + a_n b_n$

<p>جزء برنامج الفورتران</p> <p>UP1 = SQRT(INNPRO(X, X, N)) UP2 = SQRT(INNPRO(Y, Y, N)) DOWN = SQRT(INNPRO(X, Y, N)) ANGLE = UP1*UP2/DOWN</p>	<p>البرنامج الفرعي</p> <p>REAL FUNCTION INNPRO(A, B, N) DIMENSION A(N), B(N) INNPRO = 0.0 DO 100 K = 1, N INNPRO = INNPRO + A(K)*B(K) 100 CONTINUE RETURN END</p>
---	--

(لاحظ إعلان النوع حيث أن INNPRO ستكون حقيقية) .

٧-١١ اكتب برنامجاً فرعياً SUBROUTINE يطبع مجموعة متراسة ذات بعدين صفياً بصفاً (مع فرض أن المجموعة المتراسة لا تحتوي على أكثر من 13 عموداً) سيكون البرنامج بعد متغير يمكن ضبطه .

```

SUBROUTINE PRINT(A, M, N)
DIMENSION A(M, N)
DO 100 J = 1, M
WRITE(6, 10) (A(J, K), K = 1, N)
10 FORMAT(1X, 13(2X, F8.2))
100 CONTINUE
RETURN
END

```

٧-١٢ افرض A مصفوفة (L و N) مخزنة في الذاكرة . اكتب البرامج الفرعية الصغيرة SUBROUTINE التالية :

SUBROUTINE FIND(A, N, L, K, J) (1)

الذي يجد الصف J بحيث تحتوي $A(J, K)$ على القيمة الكبرى المطلقة بين $A(K, K), A(K + 1, K), \dots, A(N, K)$ (انظر مسألة ٦ - ٩) .

SUBROUTINE CHANGE(A, N, L, K, J) (ب)

الذي يستبدل عناصر الصف K في المصفوفة A مع العناصر المناظرة للصف J (انظر مسألة ٦ - ٩) .

SUBROUTINE ROWMUL(A, N, L, K, J, D) (ج)

الذي يجمع قيم الصف K إلى قيم الصف J عدد D من المرات (هذه البرامج الفرعية SUBROUTINE مستخدمة في حل المعادلات الخطية بطريقة حذف جاوس (Gauss elimination) والتي تناقش في فصل الثامن) .

SUBROUTINE FIND(A, N, L, K, J) (أ)

DIMENSION A(N, L)

J = K

KK = K + 1

DO 100 I = KK, N

IF(ABS(A(J, K)).LT.ABS(A(I, K))) J = I

100 CONTINUE

RETURN

END

SUBROUTINE CHANGE(A, N, L, K, J) (ب)

DIMENSION A(N, L)

IF(J.EQ.K) RETURN

DO 100 I = 1, L

T = A(K, I)

A(K, I) = A(J, I)

A(J, I) = T

100 CONTINUE

RETURN

END

لاحظ أننا سمحنا بإمكانية تساوي K و J ، في هذه الحالة ، ليس علينا أن نمر خلال حلقة DO التكرارية .

SUBROUTINE ROWMUL(A, N, L, K, J, D) (ج)

DIMENSION A(N, L)

DO 100 I = 1, L

A(J, I) = A(J, I) + D*A(K, I)

100 CONTINUE

RETURN

END

مسائل تكميلية

البرامج الفرعية

١٣-٧ اكتشاف الأخطاء ، إن وجدت ، في كل جملة تعريف للبرامج الفرعية التالية :

SUBROUTINE SEEK(A, B, X, M (K), Z) (ج) FUNCTION, ORDER(A, B, 2*X) (١)

SUBROUTINE LOOK(K, J(4), L, M) (د) FUNCTION(I, J, K) (ب)

١٤-٧ اكتشاف الأخطاء ، إن وجدت ، في كل جزء من برامج الفورتران التالية :

SUBROUTINE LOOK(X, M, N, Z) (ب) FUNCTION BANK(A, B, N, K) (١)
DIMENSION X(M), Y(M, K), Z(N) DIMENSION A(N, N - 1), C(K), B(200)

١٥-٧ اكتشاف الأخطاء ، إن وجدت ، في البرامج الصغيرة الفرعية التالية :

REAL SUBROUTINE LOOK(X, Y, Z) (ب)	FUNCTION DEP(X, Y, N) (١)
Z = X + Y	Z = X + FLOAT(N)
RETURN	Y = X - N*Z
END	RETURN
	END

١٦-٧ أوجد خرج البرنامج التالي :

```

JG(Δ) = K**2 + L*K + M
Δ = 3
L = 4
M = 5
I = JG(6)
J = JG(L)
WRITE(6, 10) I, J
10 FORMAT(1X, 2I10)
STOP
END
    
```

١٧-٧ (١) اكتب البرنامج الفرعي FUNCTION ADD(M, N) الذي يحسب مجموع N من الأعداد الصحيحة المتتالية ابتداء من M بالتحديد .

$$M + (M + 1) + (M + 2) + \dots + (M + N - 1)$$

على سبيل المثال $MM = ADD(5, 4)$ سيخزن الرقم : $5 + 6 + 7 + 8 = 26$ في MM

(ب) أوجد خرج البرنامج التالي الذي يستخدم البرنامج الفرعي FUNCTION السابق :

```

C      PROGRAM USING FUNCTION SUBPROGRAM
I = 1
J = 2
K = 3*I + 2*J
I = ADD(J, K - 3)
J = ADD(I - 1, J + 1)
K = ADD(K, K - 3)
WRITE(6, 10) I, J, K
10      FORMAT(1X, 3I10)
STOP
END
    
```


١٨-٧ (أ) اكتب برنامجاً فرعياً صغيراً (SUBROUTINE SUB (J,K, JSUM, JPROD, JDIFF) بحيث تعطي JSOM و JPROD و JDIFF تعطي المجموع $J + K$ وحاصل الضرب $J \cdot K$ والفرق $J - K$ على الترتيب .

(ب) أوجد خرج البرنامج التالى الذى يستخدم البرنامج الفرعى الصغير السابق :

```
C PROGRAM USING SUBROUTINE SUBPROGRAM
  J = 3
  K = 2
  CALL SUB(J, K, L, M, N)
  CALL SUB(L + M, 3*N, J, N, M)
  WRITE(6, 10) J, K, L, M, N
10 FORMAT(1X, 5(I5, 2X))
  STOP
  END
```

١٩-٧ اكتب برنامجاً حقيقياً (FUNCTION ADDS(J, K) بحسب المجموع

$$1/J + 1/(J + 1) + 1/(J + 2) + \dots + 1/K$$

عندما تكون $J \leq K$ ولكن يفتح $ADDS = 0.0$ عندما تكون $J > K$. أوجد خرج البرنامج التالى :

```
I = 2
J = 4
K = 3
X = ADDS(I, J)
Y = ADDS(J, K)
Z = ADDS(J, 5)
WRITE(6, 30) X, Y, Z
30 FORMAT(1X, 3F12.2)
  STOP
  END
```

٢٠-٧ اكتب جزء من برنامج فورتران بحساب :

$$A = x^2 + y^2 - 25$$

$$B = 4x^2 + 9y^2 - z^2$$

$$C = \sqrt{x^2 + y^2 - 36}$$

(تلجج : استخدم دالة الجملة الحساية $F(r, s, t) = r^2 + s^2 - t^2$)

برامج

٢١-٧ إذا أعطيت مجموعة متراصة $A (A_1, A_2, \dots, A_N)$ بها N عنصر ، وكان لدينا تعريف المتوسط الحسابى على أنه :

$$(A_1 + A_2 + \dots + A_N)/N$$

والمتوسط الهندسى بالآتى :

$$\sqrt[N]{A_1 \cdot A_2 \cdot \dots \cdot A_N}$$

فعليك أن :

(أ) تكتب برنامجاً فرعياً (SUBROUTINE لحساب المتوسط الحسابى والهندسى .

(ب) تكتب برنامجاً فرعياً (FUNCTION لحساب المتوسط الحسابى والهندسى .

٢٢-٧ افرض أن C مصفوفة مكونة من $(N \times N)$. اكتب البرنامج الفرعي FUNCTION الذى يجد حاصل الضرب لعناصر القطر $C(1, 1), C(2, 2), \dots, C(N, N)$

٢٣-٧ افرض أن B مصفوفة مكونة من $M \times N$:

- (أ) اكتب البرنامج الفرعي FUNCTION الذى يجد أكبر عنصر في B
 (ب) اكتب البرنامج الفرعي FUNCTION الذى يجد العنصر ذو أكبر قيمة مطلقة B .
 (ج) اكتب البرنامج الفرعي FUNCTION الذى يجد مجموع عناصر B .

٢٤-٧ إذا أعطيت أربعة امتحانات لكل طالب في فصل به 25 طالباً . وثقبت درجات كل طالب على بطاقة بالإضافة إلى ID للطالب أو للطالبة . وكانت درجة المقرر تحسب كتوسط أفضل ثلاث درجات . اكتب برنامج لحساب الدرجات الطلبة (تليخ : طريقة لحساب مجموع أفضل ثلاث درجات هي طرح أقل درجة من مجموع كل الدرجات الأربع . ومن ثم ، اكتب برنامجاً فرعياً FUNCTION لإيجاد أقل درجة ثم استخدم هذا البرنامج الفرعي في كتابة البرنامج الأساسي) .

٢٥-٧ اكتب البرنامج الفرعي الصغير SUBROUTINE في الشكل :

STAT(A, N, SUM, AVE, VAR, SD)

الذى يحسب SUM والمتوسط الحسابي AVE والتباين VAR والانحراف المعياري SD لعناصر مجموعة متارة خطية A بها N عنصراً (انظر مسألة ٦-٦)

٢٦-٧ افرض A مجموعة متراصة خطية بها N عنصراً . اكتب البرامج الفرعية الصغيرة التالية SUBROUTINE

- (أ) SORTUP(A, N) الذى يرتب A ترتيباً تصاعدياً (انظر مسألة ٦-١٢)
 (ب) SORTDN(A, N) الذى يرتب A ترتيباً تنازلياً .

٢٧-٧ اكتب البرنامج الفرعي الصغير SUBROUTINE في الشكل

DEPR(COST, N, A)

حيث تتناقص COST بطريقة مجموع الحدود على عدد N من السنوات (انظر مسألة ٥-٢٧) ، و A مجموعة متراصة خطية بها N عنصراً بحيث تدل $A(K)$ على الكمية المتناقصة في السنة K .

٢٨-٧ افرض أن J و K عددين صحيحين موجبين . اكتب البرامج الفرعية FUNCTION التالية :

- (أ) LGCD(J, K) الذى يجد القاسم المشترك الأعلى لـ J و K
 (ب) LCM(J, K) الذى يجد المضاعف المشترك الأصغر لـ J و K
 تليخ : $LCM(J, K) \cdot LGCD(J, K) = J \cdot K$.

٢٩-٧ افرض A مصفوفة $(M \times N)$ فاكتب البرامج الفرعية التالية :

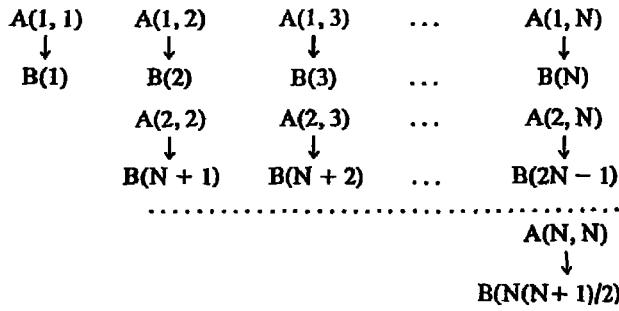
- (أ) MAXROW(A, M, N, K) الذى يحدد موضع أكبر عنصر في الصف K من المصفوفة A .
 (ب) MAXCOL(A, M, N, L) الذى يحدد موضع أكبر عنصر في العمود L من المصفوفة A .

٣٠-٧ اجعل A مصفوفة مربعة $(N \times N)$

(١) اكتب البرنامج الفرعي SYM الذي يعطى القيمة $M = 1$ إذا كانت المصفوفة A متماثلة ويعطى القيمة $M = 0$ إذا كانت غير ذلك. والمصفوفة A تكون متماثلة إذا كانت $A_{ij} = A_{ji}$ لكل من i و j .

(ب) اكتب البرنامج الفرعي $TRANS$ الذي يقبل A ثم يحورها ويعد تخزينها في A أيضاً (تُعرف المصفوفة المحورة A^T بـ A وهي المصفوفة $(N \times N)$ حيث $A_{ij} = A_{ji}$ لكل من i و j).

٣١-٧ افرض A مصفوفة $(N \times N)$ من الممكن تخزين العناصر في جزء الثلث العلوي لـ A داخل مجموعة متراصة خطية B بها $N(N+1)/2$ عنصر كما يلي :



اكتب برنامجاً فرعياً $SUBROUTINE$ لإنجاز ما سبق (هذه الطريقة يمكن بها تخزين مصفوفة متماثلة أو الثلث العلوي لمصفوفة).

اجابات للمسائل التكميلية المختارة

١٣-٧ (١) يجب ألا تكون هناك فصلة بعد $FUNCTION$ أو عند نهاية الجملة لا يمكن أن تكون $2 * X$ معامل.

(ب) البرنامج الفرعي $FUNCTION$ ليس له اسم.

(ج) لا يمكن أن تكون $M(K)$ معامل، ويجب ألا تكون هناك فصلة عند نهاية الجملة.

(د) لا يمكن أن تكون $J(4)$ معامل.

١٤-٧ (١) لا يمكن أن يكون N في جملة $DIMENSION$.

(ب) لا يمكن أن تظهر K في $Y(I)$ حيث أن K ليست معامل.

١٥-٧ (١) لم تخصص قيمة لـ DEP .

(ب) $REAL$ ليس لها معنى بما أن $LOOK$ لم تخصص لها قيمة.

١٦-٧ تطبع الأعداد الصحيحة 65 و 37 في حقول بعرض 10 .

١٧-٧ (١) حيث أن ADD لها قيمة صحيحة ، فيجب أن تعلن عنها في جملة نوع :

```

INTEGER FUNCTION ADD(M, N)
  ADD = 0
  DO 200 K = 1, N
    ADD = ADD + (M - 1) + K
200 CONTINUE
  RETURN
  END

```

(ب) الأعداد الصحيحة 14 و 42 و 34 تطبع على سطر واحد كل في حقل بعرض 10 .

١٨-٧ (١) لا نحتاج جملة نوع بما أن SUB لا تخصص له أى قيمة .

```

SUBROUTINE SUB(J, K, JSUM, JPROD, JDIF)
  JSUM = J + K
  JPROD = J*K
  JDIF = J - K
  RETURN
  END

```

(ب) ستطبع الأعداد الصحيحة 14 ، 2 ، 5 ، 8 ، 33 على سطر .

١٩-٧ تطبع الأرقام الحقيقية 1.08 و 0.00 و 0.45 في حقول بعرض 10 .

٢٠-٧

```

F(R, S, T) = R*R + S*S - T*T
A = F(X, Y, 5.0)
B = F(2.0*X, 3.0*Y, Z)
C = SQRT(F(X, Y, 6.0))

```

الفصل الثامن

أساليب البرمجة والحسابات العددية

٨ - ١ مقدمة

توجد عادة عدة طرق لكتابة برنامج للتوصل إلى حل أو حلول لمسألة معينة . وتعتمد أفضل الطرق على عدد من العوامل ، ومنها أولاً الحاجة إلى تقليل تأثير أخطاء التقريب حتى تكون النتائج دقيقة بقدر الإمكان . (انظر قسم ٢ - ١٠) - ثانياً أن يكون البرنامج على مستوى عالٍ من الكفاءة بقدر الإمكان ، حيث تعتمد كفاءة البرنامج أساساً على عنصرين :

١ - الوقت المطلوب لتشغيل البرنامج .

٢ - عدد خلايا الذاكرة المستخدمة بواسطة البرنامج .

بمعنى آخر ، يجب أن نحاول كتابة البرنامج الذي يقلل وقت الحساب وعدد خلايا الذاكرة المستخدمة بواسطة البرنامج .

وعموماً ، يمكننا كتابة برامج على مستوى عالٍ من الكفاءة ببنى عادات برمجة جيدة . فمثلاً ، إذا ظهر تعبير عدة مرات فيجب أن يخصص له اسم بحيث تحسب قيمته مرة واحدة فقط ؛ فنجد الربع إلى أس مثل $X**2.0$ ، يجب أن يكتب في الصورة $X*X$ أو $X**2$ ويجب أن يكتب

$$3.0*X**2 + 4.0*X + 5.0 \text{ بدلا من } 3*X**2 + 4*X + 5$$

ورغم أن الحاسب يمكنه أن يقبل تعبيرات ذات نمط مخطط ، وهكذا ..

إلى جانب عادات البرمجة الجيدة ، يمكن إنقاص وقت الحساب بدرجة كبيرة باستخدام النظام الحسابي (الموارزم) المناسب . على سبيل المثال إذا حسبنا قيمة متعددة الحدود :

$$a_1X^N + a_2X^{N-1} + \dots + a_NX + a_{N+1}$$

بالصورة المكتوبة لاحتاجت إلى عدد $N(N+1)/2$ من عمليات الضرب وإلى عدد N من عمليات الجمع . ولكنها ستحتاج إلى عدد N من عمليات الضرب وإلى عدد N من عمليات الجمع فقط باستخدام طريقة Horner (التي تناقص في قسم ٨ - ٦) .

يمكن أيضاً إنقاص عدد أماكن الذاكرة بدرجة كبيرة باختيار النظام الحسابي (الموارزم) المناسب . ولتوضيح ذلك افترض أننا نريد أن نعيد ترتيب عمود N من العناصر لمجموعة مترابطة خطية A في الترتيب العكسي ، أي ، بحيث تكون $A(1)$ هي الأخيرة و $A(2)$ قبل الأخيرة ، وهكذا . وأن نخزن العناصر مرة ثانية في A إحدى الطرق لتنفيذ ذلك هي أن تخصص أولاً عناصر A إلى مجموعة مترابطة أخرى B (بها N عناصر) بواسطة

$$B(N) \leftarrow A(1), \quad B(N-1) \leftarrow A(2), \quad \dots, \quad B(1) \leftarrow A(N)$$

التي يمكن أن تنجز بواسطة حلقة DO التكرارية :

```
DO 99 J = 1, N
    B(N + 1 - J) = A(J)
99 CONTINUE
```

وتكون العناصر في B في الترتيب المطلوب . من ثم ، نعيد بعد ذلك العناصر إلى A مرة ثانية وذلك بتخصيص B(1) إلى A(1) ، B(2) إلى A(2) وهكذا . لاحظ أن هذه الطريقة تطلبت عدداً إضافياً N من خلايا الذاكرة .

حقيقة يمكن أن تنجز مهمتنا باستخدام مكان ذاكرة واحد إضافي . أي أننا نستبدل A(1) ، A(N) ، A(2) ، و A(N-1) وهكذا :

```
NN = N/2
DO 88 J = 1, NN
    TEMP = A(J)
    A(J) = A(N + 1 - J)
    A(N + 1 - J) = TEMP
88 CONTINUE
```

وبذلك تكون الطريقة الثانية أكثر كفاءة من الطريقة الأولى . (مثال آخر لذلك يمكن أن نجده في مناقشتنا في قسم ٦ - ٨ حول تحريك جزء من مجموعة متراسة إلى أسفل مكان واحد) .

أخيراً ، يجب أن نشير إلى أهمية العامل البشري في عملية البرمجة . وبرغم أن كتابة 2*K في صورة K + K قد توفر مقدار لا بأس به من وقت الحاسب إلا أنها قد تجعل البرنامج أكثر صعوبة في الفهم وتسبب في ضياع الوقت البشري . ومعنى آخر ، من الأفضل أن يكون لدينا برنامج عام ومفهوم وسهل التنقل (مستقل عن الآلة) عن أن يكون لدينا برنامج معقد وغير مفهوم وملءٌ بجذعٍ معتدلة على إمكانيات الآلة .

يعالج الجزء الأول من هذا الفصل تقنيات البرمجة الفنية المتعددة لمسائل معينة تظهر مراراً ، مثل الفرز والبحث والإدماج . ويخصص الجزء الثاني من هذا الفصل لحسابات الرقية مثل حل المعادلات الخطية وإيجاد جذور متعددة الحدود وضرب المصفوفات .

٨ - ٢ الفرز

نقني بالفرز ، ترتيب العناصر في نظام ما . هذا الإجراء معتاد في الحياة اليومية فالأسماء في دليل التليفون مفروزة أبجدياً . ويمكن أن تفرز سجلات الطلبة حسب أرقام ID وهكذا ، ورغم أن الفرز يمكن أن يبدو كما لو كان مهمة تافهة . ولكن يمكن أن يكون الفرز بكفاءة صعباً جداً من الناحية العملية والنظرية .

لما كان الفرز والبحث يستخدمان في الاحتفاظ بملفات عمليات تشغيل البيانات نبدأ أولاً بتقديم بعض الاصطلاحات . يفرز ملف السجلات عند تشغيل البيانات تبعاً للمفتاح معين . على سبيل المثال ، تحتفظ الجامعة بملف لطلبتها وتسمى البيانات الخاصة بكل طالب ، في الملف بسجل . وقد يحتوي السجل على عدة حقول . على سبيل المثال ، الاسم ورقم الضمان الاجتماعي ورقم الطالب والنوع والفرقة والمتوسط التراكمي للدرجات وهكذا . عند فرز الملف تبعاً لترتيب الأبيجدى للأسماء يكون حقل الاسم هو المفتاح أما إذا كان الفرز تبعاً لرقم الضمان الاجتماعي فيكون رقم الضمان الاجتماعي هو المفتاح .

وبصدد هذا ، سندرس مشكلة فرز مجموعة متراسة من الأرقام $A(1)$ ، $A(2)$ ، ... ، $A(N)$ وليكن فرزاً تصاعدياً ، حتم يكون

$$A(1) < A(2) < A(3) < \dots < A(N)$$

سوف نذكر من أن لدينا توحيلاً فورياً للعناصر وأن هذه العناصر مخزنة في الذاكرة الأساسية ويسمى هذا فرزاً داخلياً .

أحياناً تكون الملفات كبيرة جداً بحيث نحفظ بها على وحدات التخزين الخلفية يتوصل محدود . ويسمى فرز من هذا النوع من الملفات الفرز الخارجي وهو صعب ويقع خارج حدود هذا الكتاب .

(١) الفرز الفقاعي

إحدى طرق فرز A تكون بواسطة النظام الحسابي (خوارزم) المسمى فرز فقاعي والذي تمت مناقشته في المسألة ٦ - ١ .
أى ، تقارن أولاً و $A(1)$ و $A(2)$ وترتبها في النظام المطلوب حيث يكون $A(1) < A(2)$ ثم تقارن $A(2)$ و $A(3)$ وترتبها بحيث يكون $A(2) < A(3)$ ثم تقارن $A(3)$ و $A(4)$ وترتبها بحيث يكون $A(3) < A(4)$ نكمل العمل حتى تقارن $A(N-1)$ و $A(N)$ وترتبها بحيث تكون $A(N-1) < A(N)$ بعد هذا المسح (أو المرور) خلال العناصر باكملها يقفز العنصر الأكبر إلى أعلى ، أى ، إلى المكان N .

تكرر الإجراء السابق للعناصر $A(1)$ ، $A(2)$ ، ... ، $A(N-1)$ بعد المسح (المرور) خلال هذه العناصر يقفز العنصر الأكبر إلى أعلى ، أى إلى المكان $N-1$. وهكذا نكمل هذا الإجراء ، وبعد عدد $N-1$ من دورات سيم فرز المجموعات المتراسة A تصاعدياً . تظهر خريطة سير العمليات للنظام الحسابي (الخوارزم) وترجمته للفورتان في مسألة ٦ - ١٢ .

يمكن تحسين كفاءة هذا النظام الحسابي (الخوارزم) كما يلي . فلاحظ أولاً أنه من خلال كل عملية مسح (أو مرور) (حلقة DO الخارجية) ، يحدث التبديل فقط عندما تكون هناك عناصر ليست مرتبة في النظام المطلوب . وتبعاً لذلك يمكن إضافة عداد لعدد التبديلات في كل عملية مسح (أو مرور) ، وإن لم يحدث تبديل فسوف تكون المجموعة المتراسة قد تم فرزها وتكون المهمة قد انتهت .

(ب) فرز انتقائي

يسمى النظام الحسابي (الخوارزم) الثاني للفرز ، الفرز التبدلي أو الفرز الانتقائي ويعمل كالآتي : أولاً نجد القيمة الصغرى في المجموعة المتراسة ونضعها في المكان الأول . ثم نجد القيمة الصغرى التي تليها ونضعها في المكان الثاني ، وهكذا . وفيما يلي نعرض بدقة أكثر هذا النظام الحسابي (الخوارزم) .

(١) أوجد المكان L لأصغر عنصر بين العناصر $A(1)$ ، $A(2)$ ، ... ، $A(N)$ ثم استبدل $A(L)$ مع $A(1)$

(٢) أوجد المكان L لأصغر عنصر بين $A(2)$ ، $A(3)$ ، ... ، $A(N)$ ثم استبدل $A(L)$ مع $A(2)$

... ..

($N-1$) أوجد المكان L لأصغر عنصر بين $A(N-1)$ و $A(N)$ ثم استبدل $A(L)$ مع $A(N-1)$

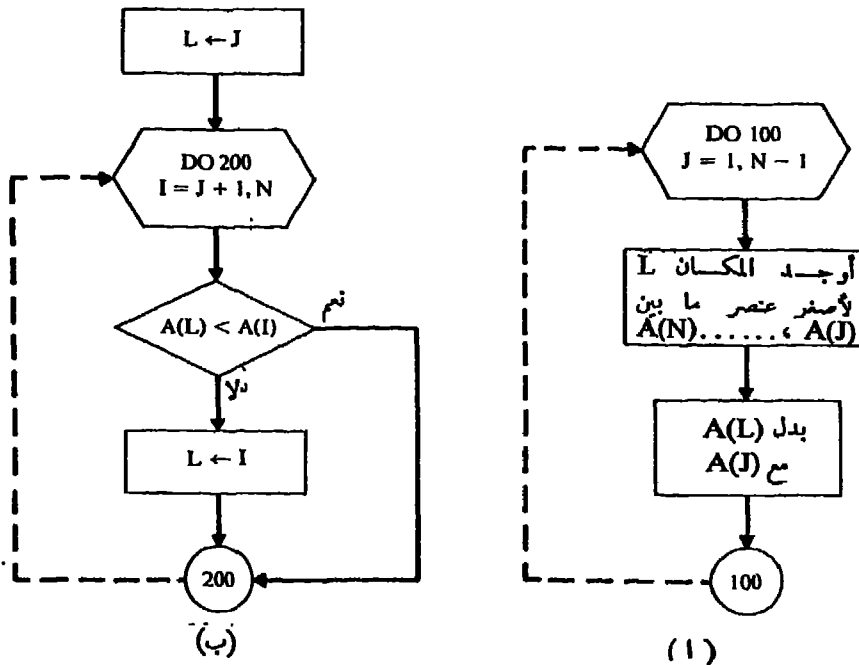
بذا تصبح المجموعة المتراسة A مفروزة في الترتيب التصاعدي .

لاحظ أن الخطوات متأللة مشابه لبعضها ، من ثم ، فنوضح أننا نستطيع أن نستخدم حلقة DO بحكومة بدليل J يتغير من $N-1$ إلى 1 . بداخل حلقة DO هذه نريد أن نجد المكان L لأصغر عنصر بين $A(J)$ ، ... ، $A(N)$ وعندئذ نستبدل $A(L)$

مع $A(J)$. تظهر خريطة سير العمليات المصغرة لخلقة DO في شكل ٨ - ١ (ب) . ينتج عن إدماج خريطة سير العمليات هذه برنامج واحد وهو برنامج الفرز المطلوب الآتي :

```

C
C   SORTING PROGRAM FOR A(1), ..., A(N)
C
C   NN = N - 1
C   DO 100 J = 1, NN
C
C       FIND LOCATION L OF SMALLEST
C
C       L = J
C       JJ = J + 1
C       DO 200 I = JJ, N
C           IF(A(L).LT.A(I)) GO TO 200
C           L = I
200      CONTINUE
C
C       INTERCHANGE A(L) WITH A(J)
C
C       T = A(L)
C       A(L) = A(J)
C       A(J) = T
100     CONTINUE
    
```



شكل ٨ - ١

ملحوظة : يمكن حساب عدد المقارنات في الفرز الفقاعي كالتالي . خلال المرور الأول نجد أن هناك عدد $(N - 1)$ من المقارنات ، وخلال المرور الثاني نجد أن هناك عدد $(N - 2)$ من المقارنات ، وهكذا . وبذلك يكون ، العدد الكلي للمقارنات هو :

$$(N - 1) + (N - 2) + \dots + 2 + 1 = \frac{N(N - 1)}{2}$$

أما بالنسبة للفرز الانتقائي ، فنحتاج إلى عدد $(N - 1)$ من المقارنات وذلك لإيجاد العنصر الأصغر . وإيجاد العنصر الأكبر الذي يليه نحتاج إلى عدد $(N - 2)$ من المقارنات ، وهكذا . لذلك يكون المجموع المطلوب من خرى هو عدد $N(N - 1)/2$ من المقارنات . نلاحظ أن متوسط عدد التباديل سوف يكون $N(N - 1)/4$ وهو نصف عدد المقارنات ، يمكن أن ننقص عدد المقارنات بواسطة التعديل المناسب (انظر مسألة ٨ - ١) . ولكن قد يبق متوسط هذه التباديل كما هو .

٨ - ٣ الإدماج

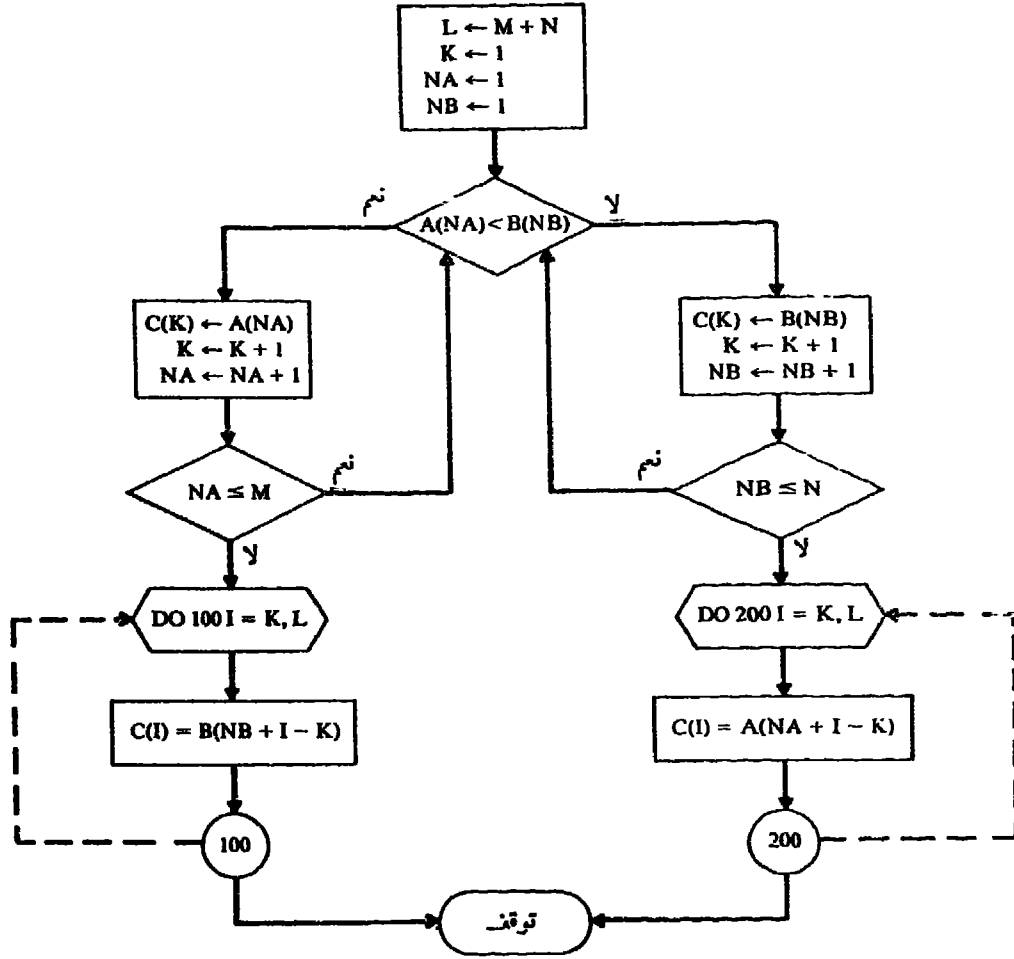
افرض أن A و B مجموعتان مترصاة مفروزة (ولتكن في الترتيب التصاعدي) بهما عدد M و N من العناصر على الترتيب . ونود أن ندسج A و B وذلك بتكوين مجموعة مترصاة مفروزة C إحدى الطرق حمل ذلك هي نقل A ، B إلى C عشوائياً ، وليكن A في البداية و B في النهاية ، ثم نفرز C كما في قسم ٨ - ٢ . لاحظ أن هذه الطريقة لا تستعمل حقيقة أن A ، B مفروزان فعلاً . وسوف نمطيك برنامج أكثر كفاءة فيما بعد .

دعنا نحفز أولاً النظام الحسابي (الخوارزم) الخاص بنا . افرض أن لدينا صفين من الطلبة مفروزان تصاعدياً حسب الطول ، وافرض : ' نريد دمجهما في صف واحد مفروز . إحدى الطرق لتنفيذ ذلك يتحقق بملء الصف الجديد تناهياً بالطالب الأقصر طولاً بين الطالبين الذين يقفان في مقدمتي الصفين الأصليين إلى أن ينتهي أحد الصفين (أى عندما لا يوجد به طلبه آخرون) . ثم ينضم الطلبة الباقون في نهاية الصف المدمج .

من أجل ترجمة هذا النظام الحسابي (الخوارزم) إلى فورتران ، يجب أن نتبع دائماً أماكن العناصر الأصغر من A ، B التي لم توضع بعد في C . سنحمل NA ، NB تشير إلى هذه الأماكن ، على الترتيب ، وبذلك عند كل خطوة نسأل السؤال التالي :

$$A(NA) < B(NB)?$$

إذا كانت الإجابة نعم ، نضع $A(NA)$ في C و نزيد NA بمقدار 1 وإذا كانت الإجابة لا ، نضع $B(NB)$ في C و نزيد NB بمقدار 1 ، إذا وصلت NA أو NB إلى قيمها العظمى (أى M أو N على الترتيب) عندئذ . نضع العناصر الباقية من المجموعة المترصاة الأخرى في C خريطة سير العمليات تظهر في شكل ٨ - ٢ .



شكل ٨ - ٢

٨ - ٤ البحث

افترض أن A مجموعة متراسة خطية مخزنة في الذاكرة ، ونريد أن نجد المكان L للمصدر D في هذه المجموعة المتراسة .

(أ) البحث المتتالي (البحث الخطي)

يمكننا أن نبحث ببساطة عن D مع عدم إعطاء أي معلومات إضافية عن A وذلك عن طريق مقارنتها بالعناصر الموجودة في A عنصراً عنصراً . تسمى هذه الطريقة بطريقة البحث المتتالي . أو البحث الخطي وقد تم توصيفها في المسألة ٦ - ١١ . من الممكن أن نرى بسهولة ، أن هذه الطريقة تتطلب عدد N من المقارنات لإيجاد D في أسوأ الحالات وعدد N/2 من المقارنات في المتوسط .

(ب) البحث الثنائي

افترض الآن أن A مجموعة متراسة مفروزة . حيث أن طريقة البحث الخطي لا تتأثر بحقيقة أن A مفروزة ، فإنه يمكن استخدام نظام حسابي (خوارزم) أكثر كفاءة ، يسمى البحث الثنائي ، وذلك البحث عن D .

دعنا نحفز أولاً النظام الحسابي الخاص بنا . افترض أننا نبحث عن اسم في دليل التليفون حيث الأسماء مرتبة أبجدياً . فنحن لا نبحث عن الاسم على التوالى من أول الصفحة إلى أن نجد المطلوب إنما نفتح الدليل من الوسط لنقرر أى نصف من الدليل يحتوي على الاسم ثم نفتح النصف من الوسط لقرر أى ربع من الدليل يحتوي على الاسم ، وهكذا إلى أن نجد الاسم المطلوب .

سنصف الآن النظام الحسابي (الخوارزم) للبحث الثنائي بصورة رسمية . أولاً اقم المجموعة المتراسة A إلى جزئين :

$$A(1), A(2), \dots, A(MID), A(MID + 1), \dots, A(N)$$

$$\text{حيث } MID = (1 + N)/2 \text{ أى :}$$

$$IBEG = 1 \quad (١)$$

$$IEND = N \quad (٢)$$

$$MID = (IBEG + IEND)/2 \quad (٣)$$

(٤) إذا كانت $D = A(MID)$ ، من ثم $L = MID$ ، وبذلك نكون قد وصلنا .

(٥) إذا كانت $D < A(MID)$ من ثم تكون D في النصف الأول من المجموعة المتراسة وإلا تكون D في

النصف الثاني من المجموعة المتراسة .

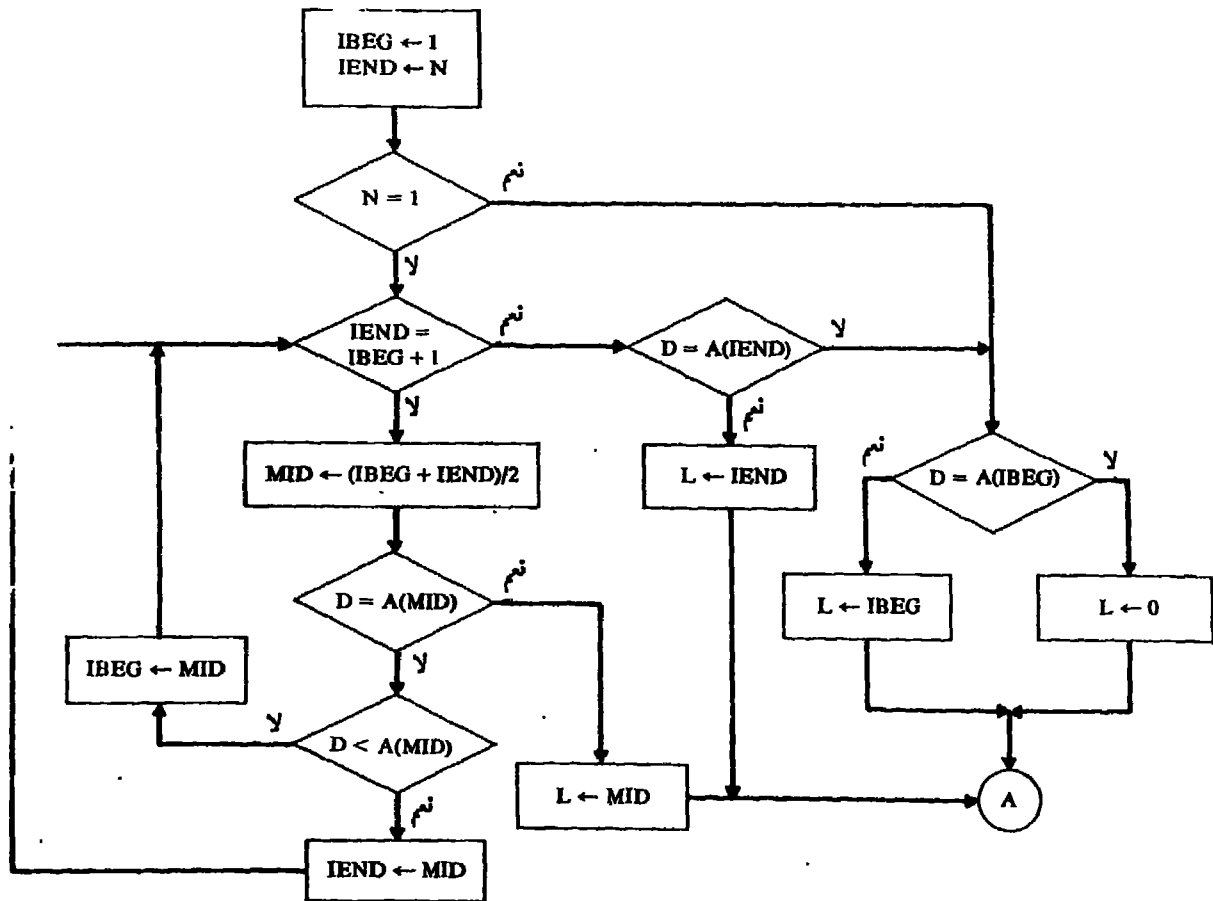
لاحظ أنه إذا كانت D في النصف الأول من المجموعة المتراسة ، من ثم تكون $A(MID)$ هي نهاية هذا النصف . لكن

إذا كانت D في النصف الثاني من المجموعة المتراسة فسوف تكون $A(MID)$ هي بداية هذا النصف . ونقش، لهذا

الفكرة حلقة تكرارية وذلك بإعادة كتابة خطوة (٥) كما يلي :

(٥) إذا كانت $D < A(MID)$ إذاً أعد وضع $IEND$ إلى MID واذهب إلى (٣) وإذا لم يتحقق ذلك فـ

ضع $IBEG$ إلى MID ، واذهب إلى (٣)



شكل ٨ - ٣

يعمل الإجراء السابق جيداً طالما يحتوي الجزء $A(\text{IBEG})$ ، ... ، $A(\text{IEND})$ من المجموعة المتراصة في أى لحظة على أكثر من عنصرين . إذا كان الجزء به عنصرين ، أى ، إذا كانت $\text{IEND} = \text{IBEG} + 1$ فسوف تكون قيمة MID هي :

$$(\text{IBEG} + \text{IBEG} + 1)/2$$

سوف تكون النتيجة IBEG مرة ثانية وذلك بسبب القسمة الصحيحة. من ثم ، سيكون لدينا حلقة تكرارية لا نهائية إذا كانت $D > A(\text{IBEG})$.
وغير صر فإننا نفحص IBEG ، IEND لئلا نرى إذا كانت أرقام صحيحة متتالية .

نقطة أخيرة . وهي إذا كانت $N = 1$ أى ، إذا ابتدأت المجموعة المتراصة أصلاً بعنصر واحد فسوف تحدث حلقة تكرارية لا نهائية أيضاً إذا كانت D لا تنتمي إلى المجموعة المتراصة وبذلك يجب أن نفحص أولاً لئلا نرى ما إذا كانت $N = 1$. تظهر خريطة سير العمليات في شكل ٨ - ٣ وترجمتها إلى الفورتران متروكة كتدريب للطلاب (مسألة ٨ - ٣) .

ملاحظة : نختم هذا القسم بمقارنة طريقة البحث الثنائي مع طريقة البحث المتتالي - باستخدام طريقة البحث الثنائي ، تحذف . ف البدائل تقريباً في كل مرة تنفذ فيها الحلقة التكرارية . (انظر شكل ٨ - ٣) . أى إذا أعطينا عدد N من الاختيارات الأصلية فسوف تبقى هناك عدد $N/2$ من الاختيارات بنهاية اللفة الأولى . وتبقى عدد $(N/2)/2$ بنهاية اللفة الثانية . وتبقى عدد $N/8 = N/(2*2*2)$ من الاختيارات بنهاية اللفة الثالثة . وهكذا . فالمدد الكلي للمقارنات في البحث الثنائي في أسوأ الحالات هو $\text{Log}_2 N$ وهو تحسن عظيم إذا قورن بالبحث المتتالي . في الحقيقة $\text{Log}_2 N$ أصغر كثيراً من $N/2$ وهو متوسط عدد المقارنات في البحث المتتالي . على سبيل المثال ، إذا كانت $N = 10,000$ فإن $N/2 = 5000$ بينما $\text{log}_2 10,000 \approx 14$

٨ - ٥ التحديث

افرض أن $A(1)$ ، $A(2)$ ، ... ، $A(N)$ مخزنة في الذاكرة ، وأن D ليست في المجموعة المتراصة A ونريد أن نضيف العنصر D إلى هذه المجموعة المتراصة .

إذا كانت عناصر A مخزنة عشوائياً ، إذن فيمكننا ببساطة إضافة D إلى نهاية المجموعة المتراصة بواسطة جملة التخصيص $A(N + 1) \leftarrow D$ أى بتخصيص D إلى $A(N + 1)$. أما إذا كانت A مجموعة متراصة مفروزة فيجب أولاً أن نجد المكان L الذى يجب أن تضاف فيه D وذلك قبل إضافة D إلى المجموعة المتراصة . وهذا يكافئ مرة أخرى نفس مهمة البحث ومع تعديلات طفيفة يمكن استخدام البحث الثنائي لإيجاد المكان L الذى تمت مناقشته سابقاً (انظر مسألة ٨ - ٦) .

هل يمكننا ببساطة تخصيص D إلى $A(L)$ بمجرد تحديد المكان L ؟ ، واضح أن الإجابة هي لا ، حيث أن هذا سيؤدى إلى إزالة $A(L)$. لذلك ، يجب أولاً أن نخل المكان بتحريك جزء المجموعة المتراصة $A(L + 1)$ ، ... ، $A(N)$ أسفل القائمة بموضع واحد :

$$A(L) \rightarrow A(L + 1) \rightarrow \dots \rightarrow A(N - 1) \rightarrow A(N) \rightarrow A(N + 1)$$

الخطوات موصوفة بالكامل في المسألة ٦ - ١٠ .

٨ - ٦ طريقة هورنر

ادرس الدالة كثيرة الحدود التالية :

$$f(x) = 3x^4 - 5x^3 + 6x^2 + 8x - 9$$

لاحظ أن هناك عدد $10 = 1 + 2 + 3 + 4$ من عمليات الضرب و 4 عمليات جمع . افترض الآن ، أننا أعدنا كتابة كثيرة الحدود بالتحليل المتال لـ X كما يلي :

$$\begin{aligned} f(x) &= (3x^3 - 5x^2 + 6x + 8)x - 9 \\ &= ((3x^2 - 5x + 6)x + 8)x - 9 \\ &= (((3x - 5)x + 6)x + 8)x - 9 \end{aligned}$$

بذلك تتطلب كثيرة الحدود 4 عمليات ضرب و 4 عمليات جمع فقط . تستخدم هذه الطريقة لإيجاد قيمة كثيرة الحدود وتسمى طريقة هورنر ، ويتم وصفها بطريقة عامة كما يلي :

ادرس الآن المعادلة العامة لكثيرة الحدود ذات الدرجة N :

$$Y = F(X) = A_1X^N + A_2X^{N-1} + \dots + A_NX + A_{N+1}$$

سوف يتطلب إيجاد قيمة كثيرة الحدود عدد مساو لـ

$$N + (N - 1) + (N - 2) + \dots + 2 + 1 = N(N - 1)/2$$

من عمليات الضرب وعدد N من عمليات الجمع . ومع ذلك إذا أوجدنا قيمة كثيرة الحدود باستخدام طريقة هورنر :

$$Y = F(X) = (((\dots((A_1X + A_2)X + A_3)X + \dots)X + A_N)X + A_{N+1}$$

فسوف نحتاج فقط إلى عدد N من عمليات الضرب وعدد N من عمليات الجمع . وبذلك تكون هذه الطريقة أكثر كفاءة .

طريقة هورنر ميزة أخرى هامة . لاحظ أنه عند كل خطوة في إيجاد قيمة Y ، نضرب في X ونجمع المعامل التالي . وبه يمكن أن نستخدم حلقة DO لإيجاد قيمة Y كما يلي :

```

Y = A(1)
NN = N + 1
DO 100 J = 2, NN
    Y = Y*X + A(J)
100 CONTINUE

```

في الحقيقة ، يمكن أن نكتب برنامج فرعي لإيجاد قيمة كثيرة الحدود من أي رتبة (انظر مسألة ٨ - ١٣) .

ملاحظة : افترض أننا جملنا $B(1)$ ، $B(2)$ ، ... ، $B(N + 1)$ تشير إلى النتائج الجزئية في طريقة هورنر بمعنى :

$$B(1) = A(1), \quad B(2) = B(1)*X + A(2), \quad \dots, \quad B(N + 1) = B(N)*X + A(N + 1)$$

وعلى ذلك يمكن أن نبين أن قيمة التفاضل $F'(X)$ هي :

$$F'(X) = B_1X^{N-1} + B_2X^{N-2} + \dots + B_{N-1}X + B_N$$

والتي يمكن أيضاً أن توجد قيمتها بواسطة طريقة هورنر . في الحقيقة جزء البرنامج التالي يعطى قيم $F'(X)$ و $F(X)$:

```

B(1) = A(1)
NN = N + 1
DO 100 J = 2, NN
    B(J) = B(J - 1)*X + A(J)
100 CONTINUE
C(1) = B(1)
DO 200 K = 2, N
    C(K) = C(K - 1)*X + B(K)
200 CONTINUE
    
```

في هذا الجزء من البرنامج $F(X) = B(N + 1)$ و $F'(X) = C(N)$

٨ - ٧ حل معادلات معينة

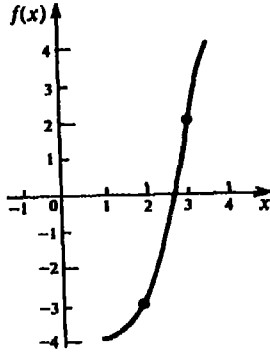
ينتشر هذا القسم إيجاد حلول حقيقية للمعادلات التي تنشأ من الدوال الكثيرة الحدود والمثلثية واللوغاريتمية ، والأسية ، وعموماً ، هناك وجهتان للمشكلة :

١- إيجاد المنطقة المجاورة للجذر .

٢- تقريب الجذر .

لأسباب تعليمية خاصة نناقش أولاً طرق تقريب الجذر إذا كانت المنطقة المجاورة معروفة .

ادرس المعادلة كثيرة الحدود التالية :



$$f(x) = x^3 - 4x^2 + 6x - 7 = 0$$

لاحظ أن $f(2) = -3$ سالبة و $f(3) = 2$ موجبة . وحيث أن f هي دالة كثيرة الحدود ورسمها البياني لا يشتمل على أى نقاط عدم استمرارية في المنحنى الخاص بها ، وعلى ذلك يجب أن يقطع الرسم البياني للدالة f محور x ما بين 2 ، 3 كما هو موضح في شكل ٨ - ٤ . والنقطة التي يقطع عندها الرسم البياني محور x هي حل حقيق للمعادلة . ونعطي هنا النظام الحسابي (الخوارزم) المعروف باسم (تصنيف المدى) والذي يحدد قيمة تقريبية لمثل هذا الجذر ، أى عدد يقل عن الجذر الحقيقي بأقل من 0.001

شكل ٨ - ٤ رسم $f(x)$ بيانياً .

(١) (تصنيف المدى)

أولاً نضع $XN = 2.0$ و $XP = 3.0$ أى XN هي النقطة التي عندها f سالبة ، و XP هي النقطة التي عندها f موجبة . نحدد النقطة الوسطى XM بين XP و XN بالمعادلة التالية :

$$XM = (XN + XP)/2$$

تحدد النقطة الوسطى XM وتوجد قيمة $f(XM)$:

$$f(XM) = f(2.5) = -1.375 \text{ و } XM = (XN + XP)/2 = 2.5$$

حيث أن $f(XM)$ سالبة ، ضع $XN = 2.5$. ثم أوجد XM و $f(XM)$ مرة أخرى :

$$f(XM) = f(2.75) = 0.046875 \text{ و } XM = 2.75$$

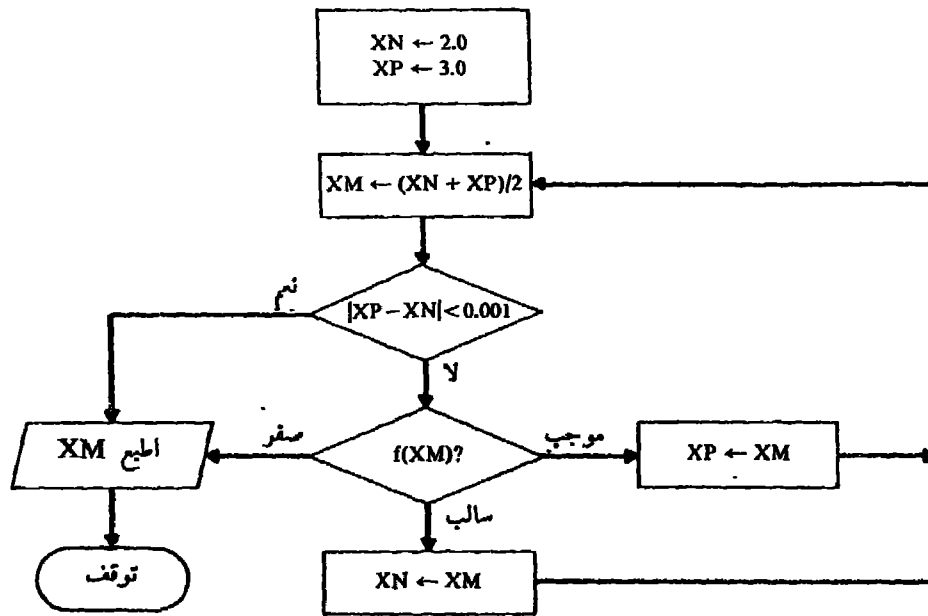
حيث أن $f(XM)$ موجبة نضع $XP = 2.75$ لاحظ أن الجذر يقع الآن بين 2.5 و 2.75

نكمل الإجراء السابق . إذا حصلنا على $f(XM) = 0$ عند أي خطوة فإن XM هو جذر . نجا عدا ذلك ، فإن XP ، XN يقتربان من بعضهما أكثر فأكثر . فإذا كانت

$$|XP - XN| < 0.001$$

فإن قيمة XM الجديدة هي قيمة مقربة للجذر الحقيقي في حدود 0.001 .

تظهر خريطة سير العمليات للنظام الحسابي (الخوارزم) في شكل ٨ - ٥ .



شكل ٨ - ٥

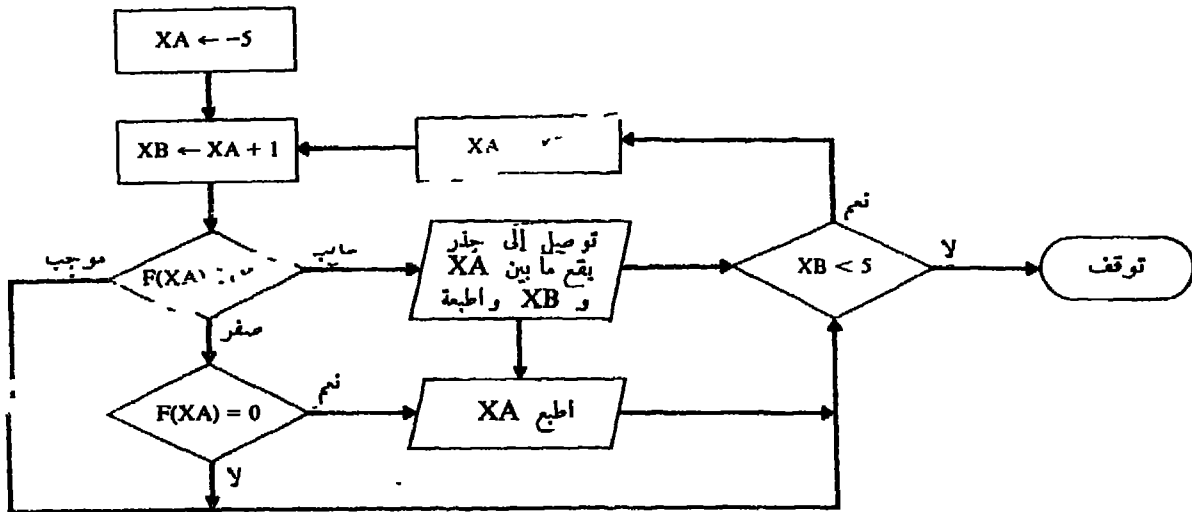
(ب) مواقع الجذور

يسمى النظام الحسابي (الخوارزم) السابق بصورة جيدة إذا أعطينا الدالة $f(x)$ والنقط A ، B التي عندها $f(A)$ و $f(B)$ لها إشارات مختلفة . ولكن المهمة الأكثر صعوبة هي إيجاد هاتين النقطتين A و B . سنناقش هذا فيما يلي :

افرض أننا اعطينا دالة $f(x)$ نبحث عن جذورها وليكن ما بين $x = -5$ و $x = 5$ يمكن أن نحدد قيمة $f(x)$ عند $5, \dots, -4, -5$ لو تغيرت إشارة الدالة f عند أي زوج من القيم K و $K + 1$ فنعرف أن $f(x)$ لها جذر بين K و $K + 1$ ويمكن أن نكل كما في (١) وبالتحديد نضع أولاً $XA = -5$ و $XB = XA + 1 = -4$ ونختبر حاصل الضرب .

$$F(XA) * F(XB)$$

إذا كان حاصل الضرب سالباً فإن إشارة $F(XA)$ و $F(XB)$ تكون مختلفة وبذا نحدد ونطبع جذر يقع ما بين XA و XB ومن ناحية أخرى ، إذا كان حاصل الضرب موجباً فإن كلا من $F(XA)$ و $F(XB)$ لها نفس الإشارة ، ولا نعرف إذا ما كان هناك جذراً يقع ما بين XA و XB أم لا . في كلتا الحالتين ، نعيد تحديد قيمة XA لتكون $XA = -4$ ثم نعيد تحديد قيمة XB عند $XB + 1 = -3$ و نكرر نفس الإجراء ينهي التكرار عند $XB = 5$. يعطى الشكل ٨ - ٦ خريطة سير العمليات لأن هذا النظام الحسابي (الخوارزم) تأخذ أيضاً خريطة سير العمليات في الاعتبار احتمال أن يكون حاصل الضرب $F(XA) * F(XB) > 0$ صفراً ، وفي تلك الحالة قد تكون XA جذراً .



شكل ٨ - ٦

من المهم أن نتذكر أن من الممكن أن تظل $f(x)$ لها جذور حقيقية رغم أن إشارتها لا تتغير عند النقط ، على سبيل المثال ، فالدالة :

$$f(x) = 8x^2 - 26x + 21$$

لها جذران يقمان بين 1 و 2 ومع ذلك فهي موجبة عند كل النقط الصحيحة . وعلى ذلك قد يتطلب الأمر إيجاد قيمة $f(x)$ باستخدام قيم أصغر أو في مدى أوسع قبل أن نياس من إيجاد الحل . وستكون أي مناقشة أعمق لهذه المشكلة واقمة خارج نطاق هذا الكتاب .

٨ - ٨ التكامل العددي

افرض دالة f موجبة في المدى $D = \{a \leq x \leq b\}$ ، والمطلوب إيجاد المساحة تحت المنحنى $y = f(x)$ ما بين $x = a$ و $x = b$ هذه المساحة مظللة في شكل ٨ - ٧ وتساوي :

$$\int_a^b f(x) dx$$

أي التكامل للدالة f من a إلى b

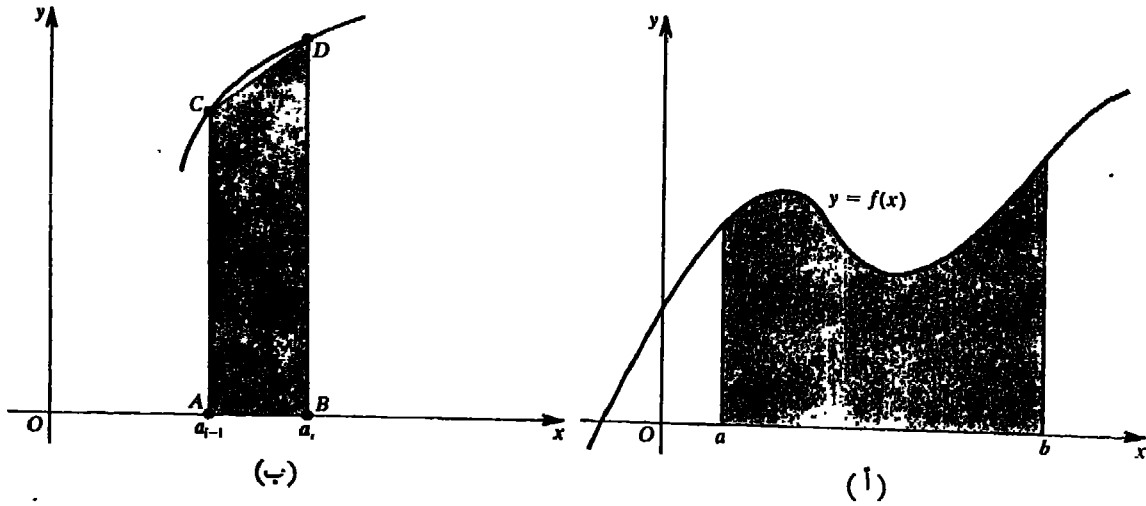
أولا تقسم D إلى عدد n من الأجزاء المتساوية كل بطول $h = (b-a)/n$ وذلك باختيار النقط :

$$a_0 = a, \quad a_1 = a + h, \quad a_2 = a + 2h, \quad \dots, \quad a_{n-1} = a + (n-1)h, \quad a_n = a + nh = b$$

ونفرض أن المعادلة الآتية تعطى تقريب للمساحة :

$$\text{area} \approx \frac{1}{2}h[f(a_0) + 2f(a_1) + 2f(a_2) + \dots + 2f(a_{n-1}) + f(a_n)]$$

تسمى هذه العلاقة « قاعدة شبه المنحرف » في ضوء مشتقاتها التي نصفها فيما يلي :



شكل ٨ - ٧

لاحظ أنه عندما تكون n كبيرة و h صغيرة ، فالمنحنى $y = f(x)$ ما بين $x = a_{i-1}$ و $x = a_i$ هو تقريباً خط مستقيم كما هو مصور في شكل ٨ - ٧ (ب) وبالتالي فالمساحة تحت المنحنى تكون أقرب إلى مساحة شبه المنحرف T_i بقروس A و B و C و D لاحظ أن :

$$AC = f(a_{i-1}) \quad \text{and} \quad BD = f(a_i)$$

وايضا h هي المسافة بين الجانبين AC و BD . من ثم :

$$\text{area}(T_i) = \frac{1}{2}h[f(a_{i-1}) + f(a_i)]$$

وتبعاً لذلك ، فالمساحة تحت المنحنى مساوية لمجموع مساحات أشباه المنحرفات تقريباً :

$$\text{area} \approx \text{area}(T_1) + \text{area}(T_2) + \dots + \text{area}(T_n)$$

وبالتالى فهي تحدد العلاقة المطلوبة .

تكون الدالة $f(x) = 3x^3 - 4x^2 + 6x + 5$ موجبة كلما كانت x موجبة . يقرأ البرنامج التالى A و B و N حيث $0 < A < B$ وكذلك يحسب المساحة التقريبية بقاعدة شبه المنحرف ، ويطلع A و B والمساحة التقريبية .

```

C      AREA BY TRAPEZOID RULE
      F(X) = ((3.0*X - 4.0)*X + 6.0)*X + 5.0
      READ(5, 10) A, B, N
10     FORMAT(2F8.2, I5)
      H = (B - A)/FLOAT(N)
      SUM = F(A) + F(B)
      NN = N - 1
      DO 200 K = 1, NN
          SUM = SUM + 2*F(A + FLOAT(K)*H)
200    CONTINUE
      AREA = H*SUM/2.0
      WRITE(6, 20) A, B, AREA
20     FORMAT(1X, 2F10.2, F15.3)
      STOP
      END
    
```

٨ - ٩ المتجهات والمصفوفات

يبحث هذا القسم في العمليات الجبرية التي تجرى على المتجهات والمصفوفات . وفي التصوير الرياضى فالتجه هو مجموعة متراصة خطية (أى ذات بعد واحد) ، والمصفوفة هى مجموعة متراصة ذات بعدين . (فى مفهوم المتجهات والمصفوفات ، يستخدم مصطلح الا متجه للأرقام المفردة) .

(١) المتجهات

تشير الرياضيات عادة إلى المتجه A (مجموعة متراصة خطية) بواسطة :

$$A = (A_1, A_2, \dots, A_n)$$

على سبيل المثال ، تشير :

$$A = (2, -3, 5, 7)$$

إلى أن A متجه به 4 عناصر هى 2 و -3 و 5 و 7

افرض A و B متجهين كل به عدد n من العناصر ، و X قيمة غير متجهة ، مجموع A و B هو المتجه الذى نحصل عليه بجمع العناصر المناظرة فى A ، B .

$$A + B = (A_1 + B_1, A_2 + B_2, \dots, A_n + B_n)$$

وحاصل ضرب المتجه A في القيمة غير المتجهة X هو المتجه الذي نحصل عليه بضرب كل عنصر من A في X .

$$X \cdot A = (XA_1, XA_2, \dots, XA_n)$$

-اصل الضرب الداخلي (نقطة غير متجهة) لـ A و B يكتب بالصورة A.B وهي القيمة غير المتجهة :

$$A \cdot B = A_1B_1 + A_2B_2 + \dots + A_nB_n = \sum_{k=1}^n A_kB_k$$

طول A (أو : norm) ، يكتب $\|A\|$ هو القيمة غير المتجهة :

$$\|A\| = \sqrt{A \cdot A} = \sqrt{A_1^2 + A_2^2 + \dots + A_n^2}$$

على سبيل المثال ، افرض $A(2, -3, 5, 7)$ و $B(4, 2, -1, 3)$ ، وعلى ذلك :

$$A + B = (2 + 4, -3 + 2, 5 - 1, 7 + 3) = (6, -1, 4, 10)$$

$$3A = (3 \cdot 2, 3 \cdot (-3), 3 \cdot 5, 3 \cdot 7) = (6, -9, 15, 21)$$

$$A \cdot B = 2 \cdot 4 + (-3) \cdot 2 + 5 \cdot (-1) + 7 \cdot 3 = 8 - 6 - 5 + 21 = 18$$

$$\|A\|^2 = 2^2 + (-3)^2 + 5^2 + 7^2 = 4 + 9 + 25 + 49 = 87, \text{ and so } \|A\| = \sqrt{87}$$

$$\|B\| = \sqrt{16 + 4 + 1 + 9} = \sqrt{30}$$

افرض A و B و C و D متجهات خطية كل بها عدد N من العناصر و X قيمة غير متجهة . فيما يلي جزء من برنامج يخزن مجموع A و B في C و يخزن حاصل الضرب غير المتجه X و A في D و يخزن الكيات غير المتجهة A.B و $\|A\|$ و $\|B\|$ في DOT و ANORM و BNORM على الترتيب :

```

DOT = 0.0
SUMA = 0.0
SUMB = 0.0
DO 100 K = 1, N
    C(K) = A(K) + B(K)
    D(K) = X*A(K)
    SUMA = SUMA + A(K)**2
    SUMB = SUMB + B(K)**2
    DOT = DOT + A(K)*B(K)
100 CONTINUE
ANORM = SQRT(SUMA)
BNORM = SQRT(SUMB)

```

(ب) جميع المصفوفات والضرب اللاموجه

تذكر أولاً أن المصفوفة هي مجموعة مترابطة ذات بعدين تكتب عناصرها عادة في شكل مستطيل حيث يشير أول دليل إلى الصف في المصفوفة والدليل الثاني إلى العمود في المصفوفة .

افرض مصفوفتين $A = (a_{ij})$ و $B = (b_{ij})$ هما عدد $m \times n$ من العناصر . وتكتب مجموع A و B بالصورة $A + B$ ، وهي المصفوفة $m \times n$ التي عناصرها ij هو $a_{ij} + b_{ij}$ أي ، نحصل عليها بجمع العناصر المناظرة في A و B . حاصل ضرب المصفوفة A والقيمة غير المتجهة k ، تكتب k.A ، وهي المصفوفة $m \times n$ التي عناصرها ij هو ka_{ij} أي ، نحصل عليها بضرب كل عنصر من A في k .

وعلى سبيل المثال ، افرض المصفوفتين A و B 2×3 هما المصفوفتان 2×3 التاليتان :

$$A = \begin{pmatrix} 1 & -3 & 5 \\ 2 & 0 & -6 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4 & 1 & -2 \\ 1 & -5 & -3 \end{pmatrix}$$

وعلى ذلك :

$$A + B = \begin{pmatrix} 1+4 & -3+1 & 5-2 \\ 2+1 & 0-5 & -6-3 \end{pmatrix} = \begin{pmatrix} 5 & -2 & 3 \\ 3 & -5 & -9 \end{pmatrix}$$

$$-3A = \begin{pmatrix} -3 \cdot 1 & -3 \cdot (-3) & -3 \cdot 5 \\ -3 \cdot 2 & -3 \cdot 0 & -3 \cdot (-6) \end{pmatrix} = \begin{pmatrix} -3 & 9 & -15 \\ -6 & 0 & 18 \end{pmatrix}$$

وكان في الجبر العادي $2A - 4B = 2A + (-4B)$ ومن م :

$$2A - 4B = \begin{pmatrix} 2 & -6 & 10 \\ 4 & 0 & -12 \end{pmatrix} + \begin{pmatrix} -16 & -4 & 8 \\ -4 & 20 & 12 \end{pmatrix} = \begin{pmatrix} -14 & -10 & 18 \\ 0 & 20 & 0 \end{pmatrix}$$

افترض الآن A و B و C و D مجموعات متراسة في صورة مصفوفة كل منها $(M \times N)$ و X و Y كيات غير متجهة. فبإيل جزء من برنامج الفورتران الذي يخزن مجموع A و B في C ويخزن $X \cdot A - Y \cdot B$ في المصفوفة D :

```
DO 100 L = 1, M
  DO 200 J = 1, N
    C(L, J) = A(L, J) + B(L, J)
    D(L, J) = X*A(L, J) + Y*B(L, J)
  200 CONTINUE
100 CONTINUE
```

لاحظ أننا استخدمنا حلقات DO المتداخلة بسبب وجود دليلين .

(ج) رمز التجميع

قبل أن نعرف ضرب المصفوفات ، سيكون من الأنسب أولاً أن نقدم رمز التجميع Σ (الحرف اللاتيني سيجمما (Sigma)).

افرض $f(k)$ هي التعبير الجبري المشتمل على المتغير k إذن فالتعبير :

$$\sum_{k=1}^n f(k) \quad \text{أو المكافئ} \quad \sum_{k=1}^n f(k)$$

له المعنى التالي : أولاً نجعل $k = 1$ في $f(k)$ لنحصل على $f(1)$

$$f(1)$$

ثم نجعل $k = 2$ في $f(k)$ لنحصل على $f(2)$ ونضيف هذه القيمة على $f(1)$ لنحصل على :

$$f(1) + f(2)$$

بعد ذلك نجعل $k = 3$ في $f(k)$ لنحصل على $f(3)$ ونضيف هذه القيمة إلى المجموع السابق ، لنحصل على :

$$f(1) + f(2) + f(3)$$

ونستمر في هذا الإجراء إلى أن نحصل على المجموع :

$$f(1) + f(2) + f(3) + \dots + f(n-1) + f(n)$$

لاحظ أننا نزيد قيمة k بمقدار 1 عند كل خطوة حتى تصل قيمة k إلى n (ونشير إلى أنه في استطاعتنا استخدام أي رمز آخر مثل i أو j بدلا من k).

ونسم أيضاً التمرير وذلك بالسماح للمجموع أن يتراوح ما بين أي رقم صحيح n_1 إلى أي رقم صحيح آخر n_2 حيث $n_1 \leq n_2$ أي نعرف :

$$\sum_{k=n_1}^{n_2} f(k) = f(n_1) + f(n_1 + 1) + f(n_1 + 2) + \dots + f(n_2)$$

وعلى سبيل المثال :

$\sum_{k=1}^5 x_k = x_1 + x_2 + x_3 + x_4 + x_5$	يكون لدينا	$f(k) = x_k$	لكل
$\sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$	يكون لدينا	$f(i) = a_i b_i$	ولكل
$\sum_{j=3}^5 j^2 = 3^2 + 4^2 + 5^2 = 9 + 16 + 25 = 50$	يكون لدينا	$f(j) = j^2$	ولكل

ويكون لدينا أيضاً :

$$\sum_{k=1}^5 (-1)^{k+1} (1/k) = 1 - 1/2 + 1/3 - 1/4 + 1/5$$

$$\sum_{k=1}^5 a_k b_k = a_{11} b_{11} + a_{12} b_{21} + a_{13} b_{31} + \dots + a_{5p} b_{p1}$$

$$\sum_{k=1}^{n+1} a_k x^{n+1-k} = a_1 x^n + a_2 x^{n-1} + \dots + a_n x + a_{n+1}$$

(د) ضرب المصفوفات

افرض $A = (a_{ij})$ مصفوفة $m \times p$ و $B = (b_{ij})$ مصفوفة $p \times n$. حاصل ضرب A و B ، يكتب بالصورة AB ، وهو المصفوفة $C = (c_{ij})$ التي لها عدد $m \times n$ من العناصر والتي يحدد عنصرها c_{ij} بواسطة :

$$c_{ij} = a_{i1} b_{1j} + a_{i2} b_{2j} + \dots + a_{ip} b_{pj} = \sum_{k=1}^p a_{ik} b_{kj}$$

(يمكن أن نرى c_{ij} على أنها قيمة غير متجهة نحصل عليها بحاصل الضرب الداخلي (نقطة) للصف i في A والعمود j في B) .

على سبيل المثال ، افرض A و B هما المصفوفتان (2×3) و (3×4) التاليتان على الترتيب :

$$B = \begin{pmatrix} 5 & 6 & 7 & 8 \\ -1 & -2 & -3 & -4 \\ 0 & 9 & 0 & 9 \end{pmatrix} \quad \text{و} \quad A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

١- البرمجة بلغة الفورترات

لنحصل على حاصل الضرب $C = AB$ نذكر أن C ستكون مصفوفة (2×4) . العنصر c_{11} هو حاصل الضرب الداخلي للصف الأول في A والعمود الأول في B :

$$c_{11} = (1, 2, 3) \cdot (5, -1, 0) = 5 - 2 + 0 = 3$$

العنصر c_{12} هو حاصل الضرب الداخلي للصف الأول في A والعمود الثاني في B :

$$c_{12} = (1, 2, 3) \cdot (6, -2, 9) = 6 - 4 + 27 = 29$$

بالمثل :

$$c_{13} = 7 - 6 + 0 = 1$$

$$c_{14} = 8 - 8 + 27 = 27$$

الصف الثاني في C نحصل عليه بضرب الصف الثاني في A بكل الأعمدة في B :

$$c_{21} = 20 - 5 + 0 = 15$$

$$c_{22} = 24 - 10 + 54 = 68$$

$$c_{23} = 28 - 15 + 0 = 13$$

$$c_{24} = 32 - 20 + 54 = 66$$

وبالتالي :

$$C = \begin{pmatrix} 3 & 29 & 1 & 27 \\ 15 & 68 & 13 & 66 \end{pmatrix}$$

لاحظ أن حاصل الضرب BA غير معروف حيث أن عدد الأعمدة في A (4) ليس مساوياً لعدد الصفوف في A (2) .

وبصورة عامة إذا كانت A مصفوفة $(M \times L)$ و B مصفوفة $(L \times N)$ فإن العنصر c_{ij} في مصفوفة حاصل الضرب $C=AB$ يمكن أن يحسب كما يلي :

```

● C(I, J) = 0.0
  DO 100 INDEX = 1, L
    C(I, J) = C(I, J) + A(I, INDEX)*B(INDEX, J)
100 CONTINUE

```

ما يلي هو في الواقع البرنامج الفرعي الكامل SUBROUTINE الذي يحسب C :

```

SUBROUTINE MATMUL(A, B, M, L, N, C)
DIMENSION A(M, L), B(L, N), C(M, N)
DO 300 I = 1, M
  DO 200 J = 1, N
    C(I, J) = 0.0
    DO 100 INDEX = 1, L
      C(I, J) = C(I, J) + A(I, INDEX)*B(INDEX, J)
100 CONTINUE
200 CONTINUE
300 CONTINUE
RETURN
END

```

٨ - ١٠ المعادلات الخطية

نكرس هذا القسم لحل نظام مكون من عدد n من المعادلات الخطية فيها عدد n من المجاهيل . أولاً ندرس الحالة الخاصة للنظام المثلي . وبعد ذلك ندرس الحالة العامة باستخدام طريقة الحذف لجاوس .

(١) النظام المثلي

يقال أن النظام التالي من المعادلات الخطية في الصيغة المثلية

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1,n-1}x_{n-1} + a_{1n}x_n &= b_1 \\ a_{22}x_2 + \dots + a_{2,n-1}x_{n-1} + a_{2n}x_n &= b_2 \\ \dots & \\ a_{n-2,n-2}x_{n-2} + a_{n-2,n-1}x_{n-1} + a_{n-2,n}x_n &= b_{n-2} \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \\ a_{nn}x_n &= b_n \end{aligned}$$

المجاهيل هنا هي x_1, x_2, \dots, x_n والمعاملات هي $a_{11}, a_{12}, \dots, a_{1n}, a_{22}, \dots, a_{2n}, \dots, a_{n-1,n-1}, a_{n-1,n}, a_{nn}$. ونفرض أن لنظام حل مفرد ؛ نظراً لعدم وجود أي حد من حدود القطر $a_{11}, a_{22}, \dots, a_{nn}$ مساوياً للصفر . وسوف نشير هنا أيضاً إلى المعادلات بواسطة L_1, L_2, \dots, L_n على الترتيب .

ولإيجاد الحل x_n في آخر معادلة L_n نحصل على :

$$x_n = b_n/a_{nn}$$

الآن ولإيجاد بقية المجاهيل نحل بالإجراء المرفق بالإحلال الخلق . وبالتحديد ، نعوض بقيمة x_n في المعادلة قبل الأخيرة L_{n-1} وذلك لإيجاد x_{n-1} فنحصل على :

$$x_{n-1} = (b_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1}$$

بعد ذلك نعوض عن x_{n-1} و x_n في المعادلة الثالثة قبل الأخيرة L_{n-2} ونحل لإيجاد x_{n-2} فنحصل على :

$$x_{n-2} = (b_{n-2} - a_{n-2,n-1}x_{n-1} - a_{n-2,n}x_n)/a_{n-2,n-2}$$

وهكذا ، أي ، لكل من $k < n$ نعوض عن x_{k+1}, \dots, x_n في L_k ونحل لإيجاد x_k فنحصل على :

$$x_k = (b_k - a_{k,k+1}x_{k+1} - \dots - a_{kn}x_n)/a_{kk} = (b_k - \sum_{m=k+1}^n a_{km}x_m)/a_{kk}$$

(اسم الإحلال الخلق يأتي من حقيقة أننا نحصل على x_k بالترتيب العكسي) .

افرض أن النظام المثلي السابق تم تخزينه في الحاسب بواسطة مجموعة متراصة $(N+1) \times N$. يخزن البرنامج الصغير الفرعي BKSUB

التالي الحل في مجموعة متراصة X :

الخطوات السابقة تحول النظام إلى الشكل التالي :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2 \\ \dots & \\ a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

(ومن الطبيعي ، لا يستلزم ذلك أن تكون a_{ij} ، b_i في النظام الجديد هي نفسها a_{ij} ، b_i في النظام الأصل) .

نكرر العملية السابقة مع النظام الفرعي E_2 و... و E_n أي ، نحذف x_2 من المعادلات E_3 و... و E_n بطريقة مشابهة للخطوتين :

١ - بدل المعادلات (فيما عدا E_1) بحيث يكون العنصر المحوري a_{22} أكبر قيمة مطلقة بين معاملات x_2 في النظام الفرعي .

٢ - اضرب E_2 في $(m_{k2} = -a_{k2}/a_{22})$ واجمعها على E_k وذلك لكل $k > 2$.

وتحول هذه الخطوات النظام إلى الشكل التالي :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + \dots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + \dots + a_{2n}x_n &= b_2 \\ a_{33}x_3 + a_{34}x_4 + \dots + a_{3n}x_n &= b_3 \\ \dots & \\ a_{n3}x_3 + a_{n4}x_4 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

بعد تكرار الاجراء السابق عدد $n-1$ من المرات ، سيتحول النظام إلى الصيغة المثالية التي يمكن أن تحل عددها بالإحلال الخلفي .

فريد الآن أن نترجم النظام الحسابي (الخوارزم) السابق إلى برنامج صغير فرعي SUBROUTINE يسمى GAUSS . سنعتبر أن نظام المعادلات الخطية مخزنة في الذاكرة بمصفوفتها المتزايدة ، أي ، المبروعة المتراسة $(N \times (N + 1))$

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1N} & A_{1,N+1} \\ A_{21} & A_{22} & \dots & A_{2N} & A_{2,N+1} \\ \dots & \dots & \dots & \dots & \dots \\ A_{N1} & A_{N2} & \dots & A_{NN} & A_{N,N+1} \end{pmatrix}$$

حيث العمود $N + 1$ للمصفوفة هو عمود الثوابت للمعادلات . برنامجنا الفرعي GAUSS سيستخدم البرامج الثلاثة الفرعية الآتية المعطاة في مسألة ٧ - ١٤ .

SUBROUTINE FIND(A, N, L, K, J) - ١

والذي يحدد الصف J بحيث يحتوي $A(J, K)$ على القيمة المطلقة بين $A(K, K)$ و $A(K + 1, K)$ و... و $A(N, K)$

SUBROUTINE CHANGE(A, N, L, K, J) - ٢

والذي يبدل عناصر الصف K مع العناصر المناظرة في الصف J في A .

SUBROUTINE ROWMUL(A, N, L, K, J, D) ٣

والذي يجمع الصف K إلى الصف J عدد D من المرات .

(في هذه البرامج الفرعية ، A مجموعة متراصة مصفوفة في صورة $N \times L$)

يمكن أن نحذف x_1 من المعادلات E_3 و E_4 و E_5 باستخدام جزء البرنامج التالي .

```
CALL FIND(A, N, N + 1, 1, J)
CALL CHANGE(A, N, N + 1, 1, J)
DO 100 I = 2, N
    CALL ROWMUL(A, N, N + 1, 1, I, -A(I, 1)/A(1, 1))
100 CONTINUE
```

بالمثل ، يمكن أن نحذف x_2 من المعادلات E_2 و E_3 و E_4 باستخدام جزء البرنامج :

```
CALL FIND(A, N, N + 1, 2, J)
CALL CHANGE(A, N, N + 1, 2, J)
DO 200 I = 3, N
    CALL ROWMUL(A, N, N + 1, 2, I, -A(I, 2)/A(2, 2))
200 CONTINUE
```

وهكذا . يستخدم البرنامج الفرعي عندما يكتمل حلقة DO لكل من الخطوات السابقة . ثم يستخدم البرنامج الفرعي BKSUB لإيجاد الحل بعد أن يكون النظام في الصيغة المثلثية .

```
SUBROUTINE GAUSS(A, N, L, X)
C
C SOLUTION BY GAUSSIAN ELIMINATION
C
DIMENSION A(N, L), X(N)
NN = N - 1
DO 99 K = 1, NN
    CALL FIND(A, N, N + 1, K, J)
    CALL CHANGE(A, N, N + 1, K, J)
    KK = K + 1
    DO 88 I = KK, N
        CALL ROWMUL(A, N, N + 1, K, I, -A(I, K)/A(K, K))
88 CONTINUE
99 CONTINUE
CALL BKSUB(A, N, N + 1, X)
RETURN
END
```

مسائل

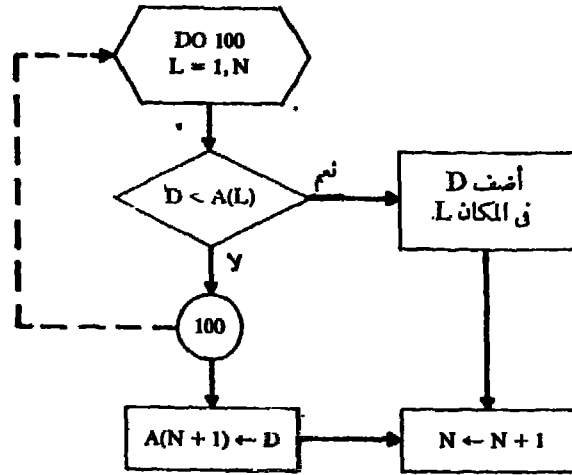
تقنية البرمجة الفنية

١ - ٨ عدل برنامج الفرز الفقاعي في مسألة ٦ - ١٢ ليحتوى العدد الذى يمد عدد مرات التبديل ، إنه البرنامج إن لم يحدث تغيير في أى عملية مسح (أو مرور) .

٢ - ٨ ترجم إلى الفورتران خريطة سير العمليات في شكل ٨ - ٢ والتي يتم بها إدماج مجموعتين متراصتين A و B مفروقتين في مجموعة متراصة C مفروزة اختر البرنامج بالبيانات التالية :

A: 1, 5, 6, 12, 14, 21; B: 2, 3, 9, 16, 18, 19, 24, 28

٢-٨ ترجم إلى الفورتان خريطة سير العمليات في شكل ٨-٢ والتي تعطى النظام الحسابي (الخوارزم) البحث الثنائي :



شكل ٨ - ٨

٤-٨ (تحديث) افترض أن $A(1)$ و $A(2)$ و \dots و $A(N)$ مفروزة في الترتيب التصاعدي ، وافترض أن D ليست في المجموعة المتراسة A . تقوم خريطة سير العمليات في شكل ٨-٨ بتحديث A باستخدام البحث الخطي لإيجاد المكان L حيث يجب أن تضاف D . ترجم خريطة سير العمليات إلى الفورتان .

٥-٨ (تحديث) عدل البرنامج في مسألة ٤-٨ بحيث يستخدم البحث الثنائي لإيجاد المكان L حيث يجب أن تضاف D .

٦-٨ (بحث وتحديث) افترض أن $A(1)$ و $A(2)$ و \dots و $A(N)$ مفروزة في الترتيب التصاعدي و D أى عنصر معطى . إذا كانت D تنتمي إلى المجموعة المتراسة A فزريد أن نجد مكانها ، لكن إذا كانت D لا تنتمي إلى المجموعة المتراسة A ، عندئذ نريد أن نقوم بتحديث A وذلك بإيجاد المكان L حيث يجب أن تضاف D .

(١) اكتب مثل هذا البرنامج باستخدام البحث المتتالي .

(ب) اكتب مثل هذا البرنامج باستخدام البحث الثنائي .

٧-٨ (الفرز بالإضافة) . يسمى نظام حسابي (خوارزم) آخر للفرز بالفرز بالإضافة ويتم وصفه فيما يلي . الفكرة الأساسية هي إضافة عنصر في مكانه المناسب بالنسبة لجزء مفروز فعلا من الملف . بالتحديد أفرز المنصرتين الأولين $A(1)$ و $A(2)$ أضف العنصر الثالث بحيث يكون $A(1)$ و $A(2)$ و $A(3)$ مرتبة بطريقة مناسبة وبصورة عامة تضاف $A(K)$ إلى جزء المجموعة المتراسة المفروز $A(1)$ و \dots و $A(K-1)$ بحيث تفرز $A(1)$ و \dots و $A(K)$ بطريقة مناسبة .

اكتب برنامجاً للفرز بالإضافة . لاحظ أن المكان الذي يجب أن تضاف فيه $A(K)$ يمكن أن يوجد بالبحث المتتالي أو البحث الثنائي . ناقش العدد الأقصى للمقارنات التي يمكن أن نحتاجها في كل حالة . ناقش عيوب الفرز بالإضافة .

٨-٨ (فرز بالإدماج) - هناك نظام حسابي (خوارزم) آخر للفرز معروف جيدا وكفاءته عالية يسمى فرز بالإدماج والخطوط

الأساسية للنظام الحسابي (الخوارزم) موضحة فيما يلي . في المرور الأول ، افرز الزوج الأول والزوج الثاني والزوج الثالث وهكذا إلى أن نفرز كل الأزواج ، أي ، نفرز كلا من :

$$\{A_1, A_2\}, \{A_3, A_4\}, \{A_5, A_6\}, \{A_7, A_8\}, \{A_9, A_{10}\}, \dots$$

في المرور الثاني يدمج الزوج الأول من عنصرين من أجزاء المجموعة المتراسة المفروزة ثم يدمج الزوج الثاني من عنصرين من أجزاء المجموعة المتراسة المفروزة وهكذا . في نهاية المرور الثاني يكون قد تم فرز كل من الرباعيات :

$$\{A_1, A_2, A_3, A_4\}, \{A_5, A_6, A_7, A_8\}, \{A_9, A_{10}, A_{11}, A_{12}\}, \dots$$

تدمج بعد ذلك أزواج من الرباعيات في المرور الثالث ، وهكذا إلى أن يتم فرز المجموعة المتراسة بأكملها . اكتب برنامجاً لتنفيذ هذه الفكرة ثم بين أن هذا النظام الحسابي (الخوارزم) يحتاج تقريباً إلى عدد $n \log_2 n$ من المقارنات .

٨ - ٩ (ا) اكتب برنامجاً صغيراً فرعياً DELETE لإلغاء جزء من مجموعة متراسة مطاة A(IN) و A(IN + 1) و A(LAST).... من المجموعة المتراسة الأصلية A .

(ب) تلميح . حرك جزء المجموعة المتراسة A(LAST + 1) و A(N).... إلى أعلى عدد مناسب من الفجوات .

(ب) : نس أن المجموعة المتراسة ID(1) و ID(2) و ID(N).... مخزنة في الذاكرة ولكن بعض المدخل أصفار . اكتب برنامجاً لضغط المجموعة المتراسة ID باستخدام DELETE

٨ - ١٠ افرض أن كل مندوب مبيعات في متجر له رقم ID . تثقب بطاقة لكل صفقة يقوم بها مندوب البيع عليها ID و كية البيع . في نهاية اليوم تجمع هذه البطاقات ، ولكن بدون ترتيب . اكتب برنامجاً لحساب إجمالي البيع الذي تم بواسطة كل مندوب بيع .

٨ - ١١ افرض أن مركز الأنشطة الحاسبية في جامعة يتقاضى من المستخدمين نتيجة خدماته \$ 600.00 لكل ساعة . ويخصص لكل مستفيد ID لكل تشغيل يتم إجراؤها على الحاسب . تثقب بطاقة برقم ID للمستفيد والوقت الذي تطلبت التشغيل بالثواني . وتكون هذه البطاقات غير مفروزة . وبفرض إضافة تكاليف إدارية قيمتها \$ 5.00 لكل تشغيل على الحاسب . اكتب برنامجاً لحساب الفاتورة الكلية لكل مستفيد . استخدم بطاقة خلفية لإنهاء البرنامج .

٨ - ١٢ في الملف الرئيسي لشركة أدلر للخدمات التليفونية يتكون كل سجل من رقم حساب العميل ، وتكلفته أو تكلفتها الأساسية الشهرية . تثقب هذه السجلات على بطاقات كل سجل على بطاقة . ويفرز الملف الرئيسي تبعاً لرقم الحساب . عند حدوث مكالمة تليفونية للمسافات الطويلة ، تثقب بطاقة برقم حساب العميل وتكلفة المكالمة وهذه البطاقات تكون غير مفروزة بأي طريقة .

افرض أن فواتير التليفون تجهز في نهاية كل شهر . باعتبار أن شركة أدلر للخدمات التليفونية لا تستخدم أكثر من 100 عميل ، وأن بطاقات تكلفة المكالمات للمسافات الطويلة تفصل عن الملف الرئيسي ببطاقة خالية . وأن المجموعة تنتهي ببطاقة خلفية . اكتب برنامجاً لحساب الفواتير لكل عميل .

الحسابات العددية

٨-١٣ اكتب برنامجاً فرعياً لدالة HORNER(A, N, X) والتي تحسب قيمة كثيرة الحدود برتبة 1-N

$$A_1X^{N-1} + A_2X^{N-2} + \dots + A_{N-1}X + A_N$$

٨-١٤ أوجد جذر الدالة $f(x) = x^3 - 4x^2 + 6x + 7$ ما بين $x = 2$ و $x = 3$. بترجمة خريطة سير العمليات الخاصة بتصنيف المدى في شكل ٨-٥ إلى فورتران .

٨-١٥ حدد وأوجد الجذور الثلاثة الحقيقية للدالة $f(x) = x^3 + x^2 - 37x + 33$ ما بين $x = -10$ ، $x = 10$ باستخدام النظام الحسابي (الخوارزم) المبين في شكل ٨-٦ وذلك لتحديد الجذور والنظام الحسابي (الخوارزم) بتصنيف المدى لأبعاد الجذور .

٨-١٦ أوجد جذرين حقيقيين للدالة $f(x) = x^2 - \sin x$

٨-١٧ أوجد المساحة تحت المنحنى $y = x^2 - 3x + 4$ ما بين $x = 3$ ، $x = 5$

٨-١٨ أوجد المساحة تحت المنحنى $y = x^2 + \sin x$ ما بين $x = 0$ ، $x = 2$

٨-١٩ اكتب برنامجاً فرعياً Subroutine (A, N, X, Y, Z) NEWTON بحسب قيمة $Z = F'(X)$ ، $Y = F(X)$ حيث

$$F(X) = A_1X^{N-1} + A_2X^{N-2} + \dots + A_{N-1}X + A_N$$

و $F'(X)$ هي تفاضل $F(X)$. قارن النتائج مع المسألة ٨-١٢ . (تلميح : انظر المناقشة في آخر قسم ٨-٦) .

٨-٢٠ (نيوتن - رابسون) ادرس المعادلة التالية :

$$F(X) = 0$$

اجعل X_0 تشير إلى قيمة في منطقة حل المعادلة . تكون المتالية :

$$X_1 = X_0 - F(X_0)/F'(X_0), \quad X_2 = X_1 - F(X_1)/F'(X_1), \quad \dots$$

،

$$X_{N+1} = X_N - F(X_N)/F'(X_N)$$

(هنا تشير $F'(X)$ التي تفاضل $F(X)$ تحت ظروف معينة ، تفصيلها تقع خارج نطاق هذا المنهج ، المتالية X_0 و X_1 و $X_2 \dots$ تقول إلى جذر $F(X)$ (تسمى هذه العملية طريقة نيوتن - رابسون) اكتب برنامجاً باستخدام هذه الطريقة لحساب $\sqrt{5}$ ، أي ، لإيجاد حل :

$$X^2 - 5 = 0$$

اختر $X_0 = 3$ وانه البرنامج عندما تصبح $|X_{N+1} - X_N| \leq 0.001$

٢١-٨ استخدم طريقة نيوتن-رابسون لإيجاد جذور $x^2 + x - 11 = 0$ إلى ثلاث خانات عشرية . أولاً اختر $x_0 = -5$ ثم $x_0 = 5$

المتجهات والمصفوفات والمعادلات الخطية

٢٢-٨ افرض أن $A = (2, -1, 0, -3)$ و $B = (1, -4, 4, 3)$ و $C = (1, 3, -2, 5)$
 (١) أوجد (١) $2A - 3B$ ، (٢) $5A - 3B - 4C$ ، (٣) $A \cdot B$ و $A \cdot C$ و $B \cdot C$ ، (٤) $\|A\|$ و $\|B\|$ و $\|C\|$
 (ب) اكتب برنامج فورتران لحساب ما سبق .

٢٣-٨ افرض

$$A = \begin{pmatrix} 1 & -1 & 2 \\ 0 & 3 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 4 & 0 & -3 \\ -1 & -2 & 3 \end{pmatrix}, \quad C = \begin{pmatrix} 2 & -3 & 0 & 1 \\ 5 & -1 & -4 & 2 \\ -1 & 0 & 0 & 3 \end{pmatrix}$$

$$D = \begin{pmatrix} 2 & 2 \\ 3 & -1 \end{pmatrix} \quad \text{و}$$

أوجد :

(a) $A + B$ (d) AC (g) D^3 (١)
 (b) $3A - 4B$ (e) BC (h) DA (٢)
 (c) $2A + 3B$ (f) D^2 (i) DB (٣)

٢٤-٨ افرض ست بطاقات كل منها تحتوي على ثلاثة أرقام حقيقية . اكتب البرنامج الذي يقرأ هذه الأرقام في صفوف المصفوفتان A و B حيث أن كل منها 3×3 ثم بعد ذلك يحسب ويطيح : (١) $5A + 2B$ (ب) $3A - 7B$ (ج) AB ،
 (د) BA ، (هـ) A^2

٢٥-٨ اعتبر المصفوفة A (6×6) بها كل عناصر القطر الرئيسي تساوى 0 وكل العناصر فوق القطر الرئيسي تساوى 1 وكل العناصر تحت القطر الرئيسي تساوى -1 كما يلي :

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & -1 & -1 & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & 0 \end{pmatrix}$$

اكتب البرنامج الذي يخزن A في الحاسب ثم يحسب ويطيح A و A^2 و A^3

٢٦-٨ أوجد حل النظام التالي :

$$\begin{aligned} 3x + 4y - 5z + 2s + 6t &= -10 \\ 5y + 7z - 4s - 6t &= -1 \\ 2z + 6s - 3t &= -35 \\ 8s + 3t &= -11 \\ 5t &= 35 \end{aligned}$$

٢٧-٨ أوجد حل النظام التالي :

$$2x + 3y + 4z - 5s + 7t = -35$$

$$8x - 2y - 3z + 9s + 3t = 53$$

$$4y + 6z - 3s - 2t = -33$$

$$5x - 7y + 8z + 3s - 9t = -19$$

$$3x + 5y - 2z + 4s + 6t = 27$$

٢٨-٨ (جاوس - جوردان) افرض نظام به عدد N من المعادلات في N من المجاهيل مخزنة في الحاسب في مجموعة متراصة A على صورة المصفوفة $(N+1) \times N$ يمكن أن نعدل طريقة جاوس ونحول المصفوفة إلى الشكل :

$$\begin{pmatrix} A_{11} & 0 & \dots & 0 & A_{1,N+1} \\ 0 & A_{22} & \dots & 0 & A_{2,N+1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & A_{NN} & A_{N,N+1} \end{pmatrix}$$

بدلاً من الصيغة الثلاثية وعلى ذلك يكون الحل :

$$X_1 = A_{1,N+1}/A_{11}, \quad X_2 = A_{2,N+1}/A_{22}, \quad \dots, \quad X_N = A_{N,N+1}/A_{NN}$$

(ويسمى هذا الإجراء طريقة جاوس - جوردان). اكتب برنامجاً لتنفيذ هذا النظام الحسابي (الخوارزم) واختبر البرنامج بالنظام المعروض في المسألة ٢٤-٨.

٢٩-٨ اجعل مصفوفة مربعة $(N \times N)$ A . وتسمى المصفوفة $(N \times N)$ B مقلوب A إذا كانت $AB = BA = I$ حيث I هي مصفوفة الوحدة أي المصفوفة التي تشتمل على آحاد على القطر الرئيسي وأسفار في الأماكن الأخرى. إذا وجد مقلوب A فهو فريد ويرمز له بالرمز A^{-1} وفيما يلي نظام حسابي (خوارزم) لإيجاد A^{-1} أولاً كون المصفوفة $(N \times 2N)$ بالصورة التالية :

$$(A, I) = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1N} & 1 & 0 & \dots & 0 \\ A_{21} & A_{22} & \dots & A_{2N} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ A_{N1} & A_{N2} & \dots & A_{NN} & 0 & 0 & \dots & 1 \end{pmatrix}$$

أي النصف الأيسر هو المصفوفة A والنصف الأيمن هو I . باستخدام تعديل لطريقة جاوس، نحول المصفوفة العليا إلى المصفوفة كما في الشكل التالي :

$$(I, B) = \begin{pmatrix} 1 & 0 & \dots & 0 & B_{11} & B_{12} & \dots & B_{1N} \\ 0 & 1 & \dots & 0 & B_{21} & B_{22} & \dots & B_{2N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & B_{N1} & B_{N2} & \dots & B_{NN} \end{pmatrix}$$

أي، حيث النصف الأيسر الآن هو I والنصف الأيمن B هو مقلوب A . اكتب برنامجاً لتنفيذ هذا النظام الحسابي (الخوارزم).

حلول المسائل المختارة

٢٢ - ٨

$$\sqrt{14}, \sqrt{42}, \text{ and } \sqrt{39} \text{ (٤)} \quad -3, -16, 12 \text{ (٣)} \quad (3, -5, 20, -44) \text{ (٢)} \quad (1, 10, 12, -15) \text{ (١)}$$

٢٣ - ٨

$$\begin{pmatrix} 26 & 18 \\ 27 & -1 \end{pmatrix} \text{ (ج)} \quad \begin{pmatrix} -5 & -2 & 4 & 5 \\ 11 & -3 & -12 & 18 \end{pmatrix} \text{ (د)} \quad \begin{pmatrix} 5 & -1 & -1 \\ -1 & 1 & 7 \end{pmatrix} \text{ (١)}$$

$$\begin{pmatrix} 2 & 4 & 12 \\ 3 & -6 & 2 \end{pmatrix} \text{ (ح)} \quad \begin{pmatrix} 11 & -12 & 0 & -5 \\ -15 & 5 & 8 & 4 \end{pmatrix} \text{ (٥)} \quad \begin{pmatrix} -13 & -3 & 18 \\ 4 & 17 & 0 \end{pmatrix} \text{ (ب)}$$

$$\begin{pmatrix} 6 & -4 & 0 \\ 13 & 2 & -12 \end{pmatrix} \text{ (ط)} \quad \begin{pmatrix} 10 & 2 \\ 3 & 7 \end{pmatrix} \text{ (٦)} \quad \begin{pmatrix} 14 & -2 & -5 \\ -3 & 0 & 17 \end{pmatrix} \text{ (ز)}$$

$$x = 3, y = -2, z = 5, s = -4, t = 7 \quad ٢٦ - ٨$$

$$x = 2, y = 1, z = -5, s = 3, t = -1 \quad ٢٧ - ٨$$

الفصل التاسع

معلومات الحروف ، المتغيرات المنطقية والعمليات الحسابية

٩ - ١ مقدمة

تم تطوير الفورتران أساساً لترجمة المادلات FORMULA TRANSLATION. ومع ذلك ، يمكن أن يتناول الفورتران أيضاً معلومات حرفية . نذكر أنه يمكننا أن نطبع رسائل بوضع الرسالة بين زوج من الفصلات العليا في جملة FORMAT . على سبيل المثال :

```
WRITE(6, 10)
10 FORMAT(IX, 'THE END')
```

ستسبب طبع الرسالة :

THE END

هذه الرسالة بها سبعة حروف T و H و E ومسافة خالية (التي نرمز إليها بواسطة b) ثم E و N و D . يسمح لنا الفورتران أيضاً بقراءة وتخزين مثل هذه السلسلة من الحروف والتعامل معها ، مثلاً ، كعملية فرز قائمة أسماء أجنبية . سوف نناقش في هذا الفصل معالجة الرموز باستخدام الفورتران .

وسوف نناقش أيضاً في هذا الفصل . استخدام الفورتران لقيام بعمليات منطقية وإيجاد قيمة متغيرات منطقية .

٩ - ٢ تخزين الحروف

عندما تخزن الحروف في ذاكرة الحاسب ، يتم تخزينها بأكواد . وفي الآلة الثنائية تخزن الحروف كسلسلة من 0's و 1's (أي اصفار وأحاد) . وواضح أنه يمكن تفسير مثل هذه السلسلة كرقم ثنائي . يطلق على هذا الرقم المكافئ المدى للمعلومات الحرفية . نقطة واحدة تحتاج إلى تأكيد وهي أن التمثيل الداخلي لـ .

1234

عند تخزينها كسلسلة حرفية يختلف تماماً في الشكل عن تمثيلها الداخلي كعدد صحيح .

هناك حد أعلى لعدد الحروف التي يمكن أن تخزن في أي خلية ذاكرة محددة . سيمس هذا الرقم ، (والذي يتغير من آلة إلى أخرى) ، سعة الحروف في الآلة (أو ببساطة سعة) وسيتمزله بالحرف M. وتكون M=4 لآلات سلسلة IBM 360/370 ، وتكون M=10 في سلسلة CDC 6000 (ولجمل البرامج قابلة للتقل من آلة إلى أخرى ننصح بأن نفترض $M = 4$) .

يمكن أن نستخدم أماكن مهيمة فقط إذ أنها تتجنب أي أخطاء تقريب قد تظهر عند تداول المعلومات الحرفية .

جملة DATA غير المنفذة ، هي إحدى الطرق لتخزين المعلومات الحرفية (نناقش جملة DATA بالتفصيل في قسم ١٠-٢) . وعلى

سبيل المثال ، الجملة :

```
DATA NAME/'PAT'/
```

والتي يجب وضعها قبل أى جملة منفذة فى البرنامج ، ستقوم بتخزين السلسلة الحرفية PAT فى المكان المسى NAME . وتمتلئ الفصلات العليا بإشارة للماسب أن PAT ستخزن كحروف ، وتستخدم الشروط المائلة لتحيط عنصر الإدخال ، وبذلك نفضلها عن اسم المتغير .

وباعتبار أن سعة الحروف $M = 4$ فسوف نخزن PAT فى الذاكرة كالتالى :

NAME	P	A	T	␣
------	---	---	---	---

أى أن السلسلة الحرفية ستخزن مضبوطة من الطرف الأيسر فى مكان الذاكرة وتنتهى بمسافات خالية على اليمين .

من ناحية أخرى ، إفرض أنك أعطيت الجملة :

DATA ID/'JOHNSON'/

أى ، سوف نخزن السلسلة الحرفية JOHNSON فى مكان الذاكرة المسى ID مضبوطة من الطرف الأيسر . ولما كانت السعة $M = 4$ ، فسوف تبتء الحروف الزائدة من على اليمين ، ويتم تخزين JOHN فقط فى ID .

ID	J	O	H	N
----	---	---	---	---

بالتجديد ، ستخزن الحروف الأربعة الأولى فقط (الحروف الأربعة من أقصى اليسار) حيث $M = 4$ ويجدر ملاحظة التعليقات الآتية :

١ - بعض الحسابات تسمح لظهور الثوابت الحرفية فى جمل منفذة . فى هذه الحالة يمكن أن نخزن المعلومات الحرفية بمباراة تخصيص .

على سبيل المثال ، الجملة :

NAME = 'PAT'

سيكون لها نفس تأثير الجملة

DATA NAME/'PAT'/

أى أننا ستخزن السلسلة الحرفية PAT بين الفصلات العليا فى موقع ذاكرة يسى NAME .

٢ - مع $M = 4$ سوف تتطلب مو تعين على الأقل لتخزين سلسلة الحروف JOHNSON على سبيل المثال ، أى ما يأتى :

DATA IDA/'JOHN'/
DATA IDB/'SON'/

DATA IDA/'JOHN'/, IDB/'SON'/

أو

DATA IDA, IDB/'JOHN', 'SON'/

والنتيجة أن JOHN ستخزن فى IDA وتخزن SON فى IDB :

IDA

J	O	H	N
---	---	---	---

 IDB

S	O	N	b
---	---	---	---

٣ - نستطيع أن نستخدم حقل H أيضاً في جملة DATA بدلا من الفصولات العليا . على سبيل المثال :

DATA NAME/3HPAT/

تخزن أيضاً سلسلة الحروف PAT و NAME (أنظر قسم ٩ - ٥) .

٤ - رغم أن بعض الحسابات لاتسمح للثوابت الحرفية أن تظهر في الجمل القابلة للتنفيذ كما في (١) فإنه يمكننا دائماً نسخ محتويات مكان إلى داخل مكان آخر بحملة تخصيص . وعلى سبيل المثال وباعتبار $M = 4$ فسوف يقوم جزء البرنامج :

DATA NAME/'PAT'/
NEXT = NAME

بتخزين PAT_b وهي محتويات NAME في مكان الذاكرة المسمى NEXT .

NEXT

P	A	T	b
---	---	---	---

لاحظ أن NEXT و NAME كلاهما متغير ثابت . ننصح بعدم مزج الأنواع في جمل التخصيص هذه .

تنبيه : عندما يتجاوز طول الثابت الحرفي l في جملة DATA سمة الحرف M تغطي بعض المترجمات إشارة خطأ . هناك احتمالات أخرى سوف يتم مناقشتها في قسم (١٠ - ٣) ولكن كل المترجمات تغطي نفس النتيجة عندما يكون $M \leq 1$.

٩ - ٣ حقل A

طريقة أخرى لتخزين المعلومات الحرفية هي قراءة المعلومات بواسطة جملة READ فتلا . عندما تقرأ المعلومات الحرفية أو تطلع ، فإننا نستخدم حقل A التي تأخذ مواصفاته الشكل :

Aw

حيث w هي عرض الحقل . (ونستطيع أن نستخدم RAW أيضاً ، حيث r هي معامل تكرار) .

(١) إدخال

افرض أن السمة $M = 4$ وتم تثقيب بطاقات بيانات كما يلي :

1	2		
1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5	
LONGFELLOW			

وعلاوة على ذلك ، افرض أمر الإدخال

READ(5, 10) NAME
10 FORMAT(A3)

بهذا تخبر الحاسب أن سلسلة الحروف التي في الأعمدة من 1 إلى 3 ستخزن في المكان المسمى NAME وهنا عرض الحقل $w = 3$ لا يتعدى سعة الحرف M ، ولذا ستخزن سلسلة الحروف مضبوطة من الطرف الأيسر في الذاكرة ومزودة بمسافة خالية على اليمين :

NAME

L	O	N	b
---	---	---	---

من ناحية أخرى ، افترض أمر الإدخال هو :

READ(5, 20) NAME
20 FORMAT(A10)

هنا يتجاوز عرض الحقل ($w = 10$) السعة M . رعل ذلك سيتم تخزين الأحرف الأربعة فقط ($M = 4$) من الحقل والموجوده على أقصى اليمين في الذاكرة :

NAME

L	L	O	W
---	---	---	---

لاحظ أن جملة DATA :

DATA NAME/'LONGFELLOW'/

سيكون لها نتيجة مختلفة . بالتحديد ، ستأمر جملة DATA بإدخال الحروف الأربعة $M = 4$ التي على أقصى اليسار فقط ويتم تخزينها في الذاكرة :

NAME

L	O	N	G
---	---	---	---

يمكن تلخيص النتائج السابقة كما يلي :

قواعد إدخال الحروف باستخدام حقل A . افترض M هي سعة الحرف و w هي عرض الحقل المعطى في جملة FORMAT للإدخال.

١ - إذا كانت $w \leq M$ تخزن الحروف w من الحقل في الذاكرة مضبوطة من جهة اليسار مع إضافة مسافات خالية بطول $M - w$ على اليمين .

٢ - إذا كانت $w > M$ تخزن الحروف M التي على أقصى اليمين من الحقل في الذاكرة .

وتبعاً لذلك فنجد قر ' LONGFELLOW باستخدام بطاقة البيانات السابقة ومع $M = 4$ نحتاج على الأقل إلى ثلاثة أماكن ذاكرة للتخزين ، فثلا :

READ(5, 10) L1, L2, L3
10 FORMAT(3A4)

ستخزن أول أربع حروف LONG في L1 والحروف الأربعة الثانية FELL في L2 والحروف الأربعة الثالثة OWbb في L3 :

L1

L	O	N	G
---	---	---	---

 L2

F	E	L	L
---	---	---	---

 L3

O	W	b	b
---	---	---	---

في حالة تغيير جملة FORMAT إلى :

10 FORMAT(2A4, A2)

تقرأ عشرة أعمدة فقط . ومع ذلك ستكون النتيجة النهائية مطابقة لأن الحروف OW تخزن مضبوطة من اليسار في L3 مع إضافة مسافتين خاليتين على اليمين .

من الواضح أننا نستطيع أن نستخدم أيضاً مجموعة متراسة لتخزين الاسم LONGFELLOW كما يلي :

```
DIMENSION L(3)
READ(5, 10) L
10 FORMAT(3A4)
```

أخيراً نؤكد أن المسافات تفهم كحروف خالية حينما نقرأ باستخدام حقل حرفي A مع أنها تفهم كأصفار باستخدام حقول رقمية .

(ب) الحسرج

نؤكد أولاً أنه إذا احتوى مكان ذاكرة على معلومات حرفية فيجب أن يحتوى وبكل دقة على عدد M من الحروف (أي ، تضاد مسافات خالية على اليسار إذا كان عدد حروف سلسلة الحروف الأصلية أقل من M) . وقبل أن نذكر بصفة رسمية القواعد التي تستخدم مع حقل A في المخرج سوف نعطي المثال التالي :

افرض السمة $M = 4$ ويحتوى المكان NAME على :

NAME	P	A	T	b
------	---	---	---	---

إذن سوف تطبع

```
WRITE(6, 20) NAME
20 FORMAT(IX, A4)
```

محتويات NAME وهي .

PAT_b

هنا يكون عرض الحقل المعطى w هو نفسه السمة $M = 4$ ومع ذلك إذا كانت جملة FORMAT هي :

```
20 FORMAT(IX, 2A)
```

حيث عرض الحقل المعطى $w = 2$ أقل من $M = 4$ فسوف يطبع فقط الحرفان اللذان على أقصى اليسار في المخزن ، وبالتحديد ستطبع :

PA

فقط .

من ناحية أخرى ، إذا كانت جملة FORMAT هي :

```
20 FORMAT(IX, A7)
```

حيث عرض الحقل المعطى $w = 7$ أكبر من $M = 4$ إذن ستطبع الحروف $M = 4$ من المخزن مضبوطة من اليمين في الحقل كما يلي :

bbbPAT_b

بمعنى آخر ، يتكون المخرج من عدد ثلاث مسافات خالية $M = 3$ متبوعة بمحتويات NAME .

النتائج السابقة يمكن تلخيصها فيما يلي :

لواعد إخراج حقول حرفية باستخدام حقل A . إفرض M هي سعة الحرف و w هي عرض الحقل المغطى في جملة FORMAT في المخرج .

١ - إذا كانت $w \leq M$ تكون نتيجة المخرج عدد w من الحروف الموجودة أقصى اليسار في المخزن ، أي سيتر عدد $M - w$ من الحروف أقصى اليمين .

٢ - إذا كانت $w > M$ ، تكون نتيجة المخرج عدد $w - M$ من المسافات الخالية متبوعة بعدد M من الحروف التي بالمخزن .

مثال ٩ - ١

نريد أن نثقب بطاقات لـ 200 اسم ثم نخزن وتطبع بحيث يبدو المخرج كما يلي :

BROWN	ERIC
EVERLING	MICHAEL
LIPSON	AUDREY
.....

بفرض أنه لا يوجد اسم (أول أو أخير) يتجاوز 15 حرفاً ، لذلك نثقب الإسم الأخير مضبغاً من اليسار في الأعمدة من 1 إلى 15 والإسم الأول مضبغاً من اليسار في الأعمدة من 17 إلى 31 . (وعلى ذلك ، يجب أن نفصل الإسم الأخير عن الإسم الأول بمسافة خالية واحدة على الأقل) .

بفرض أن السعة $M = 4$ فكل اسم يحتاج إلى ثمانية أماكن ذاكرة على الأقل لتخزين الـ 31 حرفاً المحتملة أن يتكون منها الاسم . وتبناً لذلك فإننا نستخدم مجموعة متراسة ذات بعدين (8×200) تسمى NAME . يجب أن نستخدم حلقات DO ضمنية من أجل تخزين وطباعة المتراسة صففاً بصف وفيما يلي جزء البرنامج :

```

DIMENSION NAME(200, 8)
DO 99 K = 1, 200
    READ(5, 15) (NAME(K, I), I = 1, 8)
    FORMAT(8A4)
    WRITE(6, 25) (NAME(K, I), I = 1, 8)
    FORMAT(1X, 8A4)
99 CONTINUE

```

٤ - مناقلة المعلومات الحرفية

كما سبق ذكره في قسم ٩ - ٢ ، فإن محتويات أي مكان ذاكرة هي متسلسلة من 0's و 1's (بافتراض أن لدينا آلة ثنائية) من ثم يمكن أن نقوم بالعمليات المترابطة أيضاً على المكافئ العددي لسلاسل الحروف . وبذلك يمكن أن نتداول المعلومات الحرفية كما نرى من المناقشة التالية .

(أ) الفسر:

أحياناً كثيرة ، نريد أن نفرز وحدات بأسمائها الحرفية ، أي نريد عمل قائمة بالأسماء الأبجدية . لترتيب السلاسل الحرفية أبجدياً ، يجب أن نعرف أولاً كيف تكود الحروف الأبجدية داخلياً . ولحسن الحظ ، فإن معظم المترجمات تكود الحروف الأبجدية فعلاً بالترتيب تصاعدي . أي أن ، المكافئ الرقمي للحرف A أقل من المكافئ الرقمي B . كثال ، إفرض أن IA و IB يحتويان على المعلومات الحرفية التالية:

IA	A	L	A	N	IB	D	A	L	E
----	---	---	---	---	----	---	---	---	---

إذن سيكون المكافئ الرقمي لـ ALAN المخزن في IA أقل من المكافئ الرقمي لـ DALE المخزن في IB وتبعاً لذلك فالتعبير المترابط :

IA.LT.IB

يكون صحيحاً . وبذلك نستطيع أن نستخدم الطرق المتعددة لفرز القيم الرقمية أيضاً عند فرز السلاسل الحرفية . وتستخدم المسألة ٩ - ٨ طريقة الفرز الفقاعي لفرز قائمة من الأسماء .

(ب) تحليل النص

افترض جملة S تم تثقيبها في بطاقة بيانات (تذكر أن بطاقة البيانات بها 80 عموداً) إدرس كيفية كتابة البرنامج الذي :

١ - يطبع الجملة S .

٢ - يحصى عدد المرات التي ظهر فيها الحرف E في الجملة S .

إذا أردنا أن نطبع الجملة S فقط يمكن أن نخزن أربعة حروف في مكان ذاكرة باستخدام مجموعة مَرَاصَة خَطِيئة بها 20 عنصراً كالتالي :

```
DIMENSION L(20)
READ(5, 15) L
15 FORMAT(20A4)
WRITE(6, 25) L
25 FORMAT(1X, 20A4)
```

ومع ذلك لا نستطيع أن نحصى عدد المرات التي ظهر فيها الحرف E بواسطة هذا البرنامج ولكن نفضل ذلك يجب أن نخزن حرفاً واحداً في كل مكان باستخدام مجموعة مَرَاصَة خَطِيئة بها 80 عنصر كما في البرنامج التالي :

```
DIMENSION L(80)
DATA JJ,'E'
READ(5, 15) L
15 FORMAT(80A1)
WRITE(6, 25) L
25 FORMAT(1X, 80A1)
N = 0
DO 99 K = 1, 80
    IF(L(K).EQ.JJ) N = N + 1
99 CONTINUE
WRITE(6, 35) N
35 FORMAT(1X, 'THE LETTER E APPEARS', 1X, I2, 1X, 'TIMES')
```

لاحظ أننا نستخدم جملة DATA لتخزين الحرف E في مكان يسمى JJ ثم نستخدم حلقة DO لمقارنة JJ مع كل حرف في L أي ، مع كل حرف في الجملة S .

٩ - ٥ حقل H

نذكر من قسم ٣-٨ أننا نستطيع أن نطبع رسائل في المخرج الخاص بنا بإحاطة سلسلة الحروف المطلوبة فصالات عليها في جملة الـ FORMAT الخاصة بالإخراج . على سبيل المثال ، افترض أن X تحتوي على 123.45 إذن فزوج الجمل WRITE-FORMAT التالي .

```
WRITE(6, 10) X
10 FORMAT(1X, 'THE BALANCE IS $', F8.2)
```

سيتم :

THE BALANCE IS \$ 123.45

بالمثل :

WRITE(6, 20)
20 FORMAT('1', 'ACCOUNT NUMBER', 10X, 'BALANCE')

سيطبع العنوان التالي على صفحة جديدة كما يلي :

ACCOUNT NUMBER BALANCE

نستطيع أيضاً أن نطبع رسائل باستخدام حقل H (حقل Hollerith) بالمواصفات التالية :

WH متبوعة بعدد W من الحروف المراد طباعتها .

يمكن تنقيح جمل FORMAT السابقة بالصورة المكافئة التالية :

10 FORMAT(1X, 16H THE BALANCE IS \$, F8.2)
20 FORMAT(1H1, 14H ACCOUNT NUMBER, 10X, 7H BALANCE)

وهنا نؤكد أن عدد الحروف W لا تحاط بفواصل عليا (إلا إذا كان المطلوب طباعة الفواصل نفسها). نذكر فيما يلي مواصفات حقل H - ومواصفات الحقل الحرفي المكافئ :

حقل الحروف	حقل H
'SEQUENCE'	8HSEQUENCE
'THE END'	7H THE END
'NOW IS THE TIME'	15H NOW IS THE TIME

ويجب مواصفات حقل H أن المسافات يجب أن تمد بدقة . ويجدر أن نلاحظ أن المسافات بداخل حقل H لا تم . حيث أن المسافة تحسب كحرف .

يمكن أيضاً استخدام حقل H للإدخال فنستطيع أن نستخدم حقل H لتحديد سلسلة حروف في جملة DATA . فمثلاً :

DATA I/3HPAT/, J/4HMARK/

لما نفس تأثير :

DATA I/'PAT'/, J/'MARK'/

نستطيع أيضاً أن نستخدم حقل H . جملة FORMAT للإدخال في مثل هذه الحالة يجب أن تظهر حروف زائفة (أو : مسافات خالية في جملة FORMAT . وتوضح هذه الحالة في المثال التالي) :

مثل ٩ - ٢

إفرض أن أول بطاقتين البيانات تم تنقيحها كالتالي :

	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
البطاقة الأولى	ANDERSON					JOHN														
البطاقة الثانية	BERGER					AUDREY														

ويتم تنفيذ جزء البرنامج التالي :

```
DO 100 K = 1, 2
  READ(5, 10)
  10  FORMAT(5X, 15HAAAAAABBBBBBCCCCC, 15HXXXXXXYYYYYYZZZZZ)
  WRITE(6, 10)
100 CONTINUE
```

عندما تنفذ جملة READ أول مرة ، تستبدل السلاسل الحرفية في الأعمدة من 6 إلى 20 والأعمدة من 21 إلى 35 بالحروف المذكورة في مواصفات حقول H في جملة FORMAT ، أي تحوّل جملة FORMAT فعلياً إلى :

```
10  FORMAT(5X, 15HANDERSON      , 15HJOHN      )
```

من ثمّ عندما تنفذ جملة WRITE يطبع الإسم الموجود على بطاقة البيانات . بالمثل عندما تنفذ جملة READ للمرة الثانية ، يقرأ الإسم على بطاقة البيانات الثانية من خلال جملة FORMAT وسوف تطبع عندما تنفذ جملة WRITE وتبعاً لذلك سيبدو الخرج كما يلي :

```
ANDERSON      JOHN
BERGER        AUDREY
```

٩ - ٦ ثوابت منطقية ومتغيرات منطقية

خلال مناقشة جملة IF المنطقية في الفصل الرابع قدمنا التعبيرات المترابطة التي هي أبسط شكل للتعبيرات المنطقية . والتعبيرات المترابطة هي تعبيرات تربط تعبيرين حسابيين بإحدى أدوات الترابط الست التالية :

```
.EQ. .NE. .LT. .LE. .GT. .GE.
```

وتسمى التعبيرات المترابطة أيضاً التعبيرات المنطقية لأن كلا منها تعتبر قيمة منطقية TRUE أو FALSE ويمثل هذان الثابتان المنطقيان في الفورتران بالتالي :

```
.FALSE.      و      .TRUE.
```

على الترتيب . ونؤكد أن الكلمات TRUE و FALSE يجب أن يسبق ويقب كل منها نقطة .

تسمى أماكن التخزين (المتغيرات) التي تستخدم لتخزين ثوابت منطقية متغيرات منطقية . ومثل كل المتغيرات يتكون المتغير المنطقي من واحد إلى ستة حروف أبجدية رقمية ويجب أن يكون أول حرف أبجدي كما يجب أن تملأ أسماء المتغيرات المنطقية بجملة النوع LOGICAL فشلا الجملة :

```
LOGICAL I, A45X, CAR
```

تعلن أن I و A45X و CAR متغيرات منطقية . يمكن أيضاً أن نعلن عن مجموعات مترابطة منطقية بجملة LOGICAL فتلا :

```
LOGICAL A(15)
```

لا تملأ فقط أن A مجموعة مترابطة منطقية ، ولكنها تجزأ أيضاً 15 مكاناً في الذاكرة باسم A . وحيث أن جملة LOGICAL غير منفذة ، لذا يجب أن تظهر قبل أي جملة منفذة في البرنامج .

٩ - ٧ العمليات والتعبيرات المنطقية

لستطيع أن نبني التعبيرات المنطقية الأكثر تعقيداً من التعبيرات المترابطة في الفورتران بواسطة المعاملات الثلاث الآتية (الموصلات) :

.AND.
.OR.
.NOT.

ومرة ثانية نؤكد على أن الكلمات AND و OR و NOT تسبقها وتبناها نقطة .

إذا أعطيت أى تمسرين متعقبتين $lexp1$ و $lexp2$ ، فمن الممكن أن تكون التعبير المنطقى المركب التالى :

$lexp1.AND.lexp2$
 $lexp1.OR.lexp2$
 $.NOT.lexp1$

تعتمد القيم المنطقية لهذه التعبيرات المركبة على القيم المنطقية لكل من $lexp1$ و $lexp2$ وهى تظهر فى الجداول ٩-١ و ٩-٢ (حيث T تعنى تحقق و F لايتحقق) . لاحظ أن :

١ - المعامل . AND . يتحقق عندما يتحقق $lexp1$ و $lexp2$ معاً .

٢ - المعامل . OR . لايتحقق عندما لايتحقق $lexp1$ و $lexp2$ معاً .

٣ - المعامل . NOT . هى أداة أحادية وهى تغير القيمة المنطقية لأى تعبير منطقى من T إلى F أو من F إلى T .

هذه المعاملات مثلها مثل الروابط المستخدمة فى اللغة الإنجليزية لتكوين الجمل المركبة .

جدول ٩ - ١

$lexp1$	$lexp2$	$lexp1.AND.lexp2$	$lexp1.OR.lexp2$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

جدول ٩ - ٢

$lexp1$	$.NOT.lexp1$
T	F
F	T

مثال ٩ - ٣

(أ) إدرس التعبيرات المنطقية الآتية :

(I.LT.J).OR.(K.GT.10)	(١)
(X.GT.1.0).AND.(K.LT.(I + 10))	(٢)
.NOT.(X.GT.0.0)	(٣)

ويصبح لدينا التالي :

يتحقق (١) إذا كانت $I < J$ أو $K > 10$.

يتحقق (٢) إذا كان كل من $X > 1$ و $K < I + 10$.

يتحقق (٣) إذا لم يتحقق $X > 0$ أى إذا كانت $X \leq 0$.

(ب) يقع الطلبة المبتدئون في غلطة شائعة وهي ترجمة $2 \leq x \leq 1$ إلى الفورتران بكتابة :

1.0.LE.X.LE.2.0

يجب أن نذكر أن $2 \leq x \leq 1$ تبنى $x \leq 1$ و $x \leq 2$. ومن ثم الفورتران الصواب المكافئ هو :

(1.0.LE.X).AND.(X.LE.2.0)

رغم أن التعبيرات المنطقية تستخدم كثيراً في جمل IF المنطقية ، يمكن أيضاً أن تفسر وتعمل مسائل الجبر البولي والمنطق الاتراخي باستخدام هذه التعبيرات .

٩ - ٨ تخصيص القيم المنطقية

يمكننا أن نخزن ثوابت منطقية في متغيرات منطقية بجملة تخصيص . فمثلاً :

LOGICAL A
A = .TRUE.

لتخصيص الثابت . TRUE ، إلى مكان الذاكرة المسمى A ، عموماً ، جملة التخصيص المنطقية لها الشكل التالي :

تعبير منطقي = متغير منطقي

لذلك فإن :

LOGICAL B, C
B = X.LT.Y
C = B

لتخصيص القيمة TRUE ، أو FALSE ، إلى B لهما تكون $X \ll Y$ أو لم تتحقق وبعد ذلك لتخصيص لهما B هذا إلى C .

فمثلاً :

LOGICAL D
D = (R.GT.S).OR.(X.LE.Y)

لتخصيص القيمة TRUE ، إلى D إذا تحققت $R > S$ أو $X \leq Y$ (أو تحقق كلاهما) ولتخصيص القيمة FALSE ، إلى D لهما عدا ذلك ، (لاحظ أنه من الممكن أن تستخدم الأقواس بالطريقة التقليدية) .

يمكن أيضاً أن تستخدم جملة DATA لتخصيص قيمة منطقية . مثلا :

```
LOGICAL A, B
DATA A/.TRUE./, B/.FALSE./
```

تخصص القيمة .TRUE. إلى A و .FALSE. إلى B أثناء الترجمة (نائش جملة DATA بالتفصيل في قسم ١٠ - ٢) .

يمكن أن تكون القيمة المزاملة لأسم البرنامج سرعى FUNCTION منطقية . وتكون نتيجة البرنامج الفرعى FUNCTION القيمة .TRUE. فقط إذا كانت X عنصراً في المجموعة المتراسة A .

```
LOGICAL FUNCTION CHECK(A, N, X)
DIMENSION A(N)
CHECK = .TRUE.
DO 99 I = 1, N
    IF(X.EQ.A(I)) RETURN
99 CONTINUE
CHECK = .FALSE.
RETURN
END
```

لاحظ إعلان النوع في جملة تعريف FUNCTION الذى يخبر الحاسب أن CHECK متغير منطقي وليس متغيراً حقيقياً .

٩ - ٩ حقل L

يمكن أن نخزن أو نطبع المتغيرات المنطقية بجمل READ و WRITE . ويتم هذا عن طريق حقل L الذى صيغته لها المواصفات المبينة في الشكل التالي :

Lw

حيث تشير w إلى عرض الحقل . (يمكن أن نستخدم أيضاً rLw ، حيث r هى معامل التكرار) ويستخدم حقل L كما يلي :

(أ) إدخال

يفرض أن صيغة الكود Lw تناظر المتغير المنطقى A . وعلى ذلك تخصصن إلى A القيمة .TRUE. إذا كان أول حرف غير خال في الحقل المشار إليه هو حرف T . إما إذا كان أول حرف خال هو F أو أن الحقل يأكله يتكون من مسافات فنخصص .FALSE. إلى A . (تعتبر بعض مترجمات الفورتران أى رمز أمامى غير T أو F هو .FALSE. في حين ستعطى مترجمات أخرى رسالة خطأ) .

(ب) الإخراج

في المخرج سيطلع الحرف T أو F مضبوطة من اليمين في الحقل الخاص بها تبعاً للقيمة التى يأخذها المتغير المنطقى المناظر .TRUE. أو .FALSE. كئال أنرض A و B متغيرين منطقيين يحتويان .TRUE. و .FALSE. على الترتيب . وعلى ذلك فأمر المخرج التالى:

```
WRITE(6, 10) A, B
10 FORMAT(1X, L5, 2X, L3)
```

سيطلع T في عمود 5 و F في عمود 10 . بذلك فإن كود الصيغة Lw سيطلع عدد 1 - w من الفراغات قبل طباعة الحرف T أو F .

مثال ٩ - ٤

(أ) افترض بطاقة بيانات مثقبة كالتالي :

```

      1                2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
THE FIRST DAY

```

بالإضافة إلى ذلك ، افترض أمر الإدخال هو :

```

LOGICAL A, B, C
READ(5, 10) A, B, C
10 FORMAT(L3, 2L5)

```

عروض الحقول 3 و 5 و 5 على الترتيب وأول حرف غير خال في الأعمدة من 1 إلى 3 هو T وفي الأعمدة 4 إلى 8 هو F وفي الأعمدة من 9 إلى 13 هو T . من ثم . تخصص .TRUE. ، إلى المتغيرات المنطقية A و C وتخصص .FALSE. إلى B (ب) افترض جملة FORMAT في جزء (أ) تم تغييرها إلى :

10 FORMAT(3L5)

هنا T و I و D هي أول حروف غير خالية في الحقول الثلاثة . من ثم ستخصص بعض المترجمات .TRUE. إلى A و .FALSE. إلى B, C ولكن ستعطي مترجمات أخرى رسالة خطأ .
(ـ) أوجد المخرج للجزء التالي من البرنامج :

```

LOGICAL X, Y, Z
J = 5
X = .FALSE.
Y = 3*.J.GT.12
Z = Y.AND.X
WRITE(6, 20) X, Y, Z
20 FORMAT(1X, 3L10)

```

تخصص .FALSE. إلى X ، بواسطة السطر الثالث . حيث تتحقق $12 > -15 = 3 * J$ وتخصص .TRUE. إلى Y بواسطة السطر الرابع . ورغم أن Y تتحقق و X لا تتحقق فإنه يتم تخصيص .FALSE. إلى Z بواسطة السطر الخامس . وعلى ذلك تطبع F و T و F في الأعمدة 10 و 20 و 30 على الترتيب .

٩ - ١٠ التدرج الهرمي للعمليات الحسابية

حيث أن الأدوات المنطقية تشمل تعبيرات مترابطة والتي بدورها تشمل تعبيرات حسابية ومعاملات حسابية ، فن المهم أن نفهم التدرج الهرمي الكامل للعمليات . وفيما يلي الترتيب الذي تجرى به العمليات :

١ - العمليات الحسابية

(أ) الأس (* *).

(ب) ضرب وقسمة (/ *).

(ـ) جمع وطرح (+ و -).

٢ - معاملات الترابط .

٣ - معاملات منطقية :

(أ) .NOT.

(ب) .AND.

(ج) .OR.

(وكما ذكرنا في قسم ٢ - ٦ ، سنعمل الجمع والترحح الأحادي على نفس مستوى الجمع والترحح الثنائي) .

نلاحظ أن العمليات الحسابية تنفذ أولاً ، تجري العمليات المترابطة قبل العمليات المنطقية . أما بين العمليات الست المترابطة ، فليس هناك أولويات . وبالتحديد تنفذ معاملات الترابط بالترتيب التي تظهر به من اليسار إلى اليمين . وكما في التعبيرات الحسابية ، فإنه من الممكن أن نغير هذا التدرج الهرمي باستخدام الأقواس . وعلاوة على ذلك ، فإننا ننصح بشدة باستخدام الأقواس كلما كان هذا الاستعمال يسهل فهم ترتيب عمليات . فإتلا ننصح بكتابة :

(K.GT.10).OR.((X.GE.0.2).AND.(Y.EQ.3.0)) (M.GE.1).OR.(X.NE.Y)

بدلاً من

K.GT.10.OR.X.GE.0.2.AND.Y.EQ.3.0 و M.GE.1.OR.X.NE.Y

رغم أنها متكافئة .

مسائل محلولة

بيانات حرفية

٩ - ١ باعتبار السمة الحرفية هي $M=4$ أوجد البيانات المخصصة لكل من I و J بعد تنفيذ الجملة

DATA I/'NO'/, J/'THE END'/

تخزن الـ NO مضبوطة من اليسار في I مع عدد $2 = 4 - 2$ من المسافات مضافة على اليمين ، أما في J فيخزن فقط عدد ٤ حروف من ($M = 4$) أقصى اليسار (بما فيها المسافات) من السلسلة THE END :

I	N	O		
---	---	---	--	--

J	T	H	E	
---	---	---	---	--

٩ - ٢ افترض أن $M = 4$ وإن بطاقة بيانات مثقبة كالتالي :

1	2
1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
JOHN PAUL JONES	

أوجد البيانات في المخزن إذا كان أمر الإدخال كالتالي :

```

DIMENSION K(4)      (-)          READ(5, 10) I   (١)
READ(5, 30) K       10 FORMAT(A3)
30 FORMAT(4A4)
DIMENSION J(3)      (د)          READ(5, 20) L   (ب)
READ(5, 40) J       20 FORMAT(A7)
40 FORMAT(3A5)

```

- (أ) حيث أن عرض الحقل هو ٣ فتخصص لـ I سلسلة الحروف JOH الموجودة في الأعمدة من 1 إلى 3 . وسوف تخزن في I مضبوطة من اليسار مع إضافة مسافة واحدة على اليمين ، كما في الشكل ٩ - ١ (أ) .
- (ب) حيث أن عرض الحقل هو 7 فتخصص سلسلة الحروف JOHNPA الموجودة في الأعمدة من 1 إلى 7 إلى L . وسوف تبتأ من اليسار وسيخزن في L الحروف الأربعة الموجودة أقصى اليمين فقط ، كما في الشكل ٩ - ١ (ب) .
- (ج) حيث أن عرض كل حقل هو 4 فتخزن البيانات كما في الشكل ٩ - ١ (ج) .
- (د) حيث أن عرض كل حقل هو 5 . فتخصص السلاسل 'JOHN_٥' و 'PAUL_٥' و 'JONES' إلى عناصر ل . ومع ذلك حيث أنه يوضع في التخزين الحروف الأربعة فقط الموجودة أقصى اليمين ، فتخزن البيانات كما في الشكل ٩ - ١ (د) .

J(1)	O	H	N	␣
J(2)	A	U	L	␣
J(3)	O	N	E	S

(د)

K(1)	J	O	H	N
K(2)	␣	P	A	U
K(3)	L	␣	J	O
K(4)	N	E	S	␣

(ج)

I	J	O	H	␣
---	---	---	---	---

(أ)

L	N	␣	P	A
---	---	---	---	---

(ب)

شكل ٩ - ١

٩ - ٣ افرض أن $M = 4$ وتم تقيب بطاقة كالاتي :

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
ERICSON MORTY																								

(أ) أوجد البيانات في المخزن إذا كان أمر الإدخال هو :

```

INTEGER A, B, C, D
READ(5, 10) A, B, C, D
10 FORMAT(4A4)

```

(ب) أوجد المخرج إذا كان أمر الإخراج هو :

```

WRITE(6, 30) C, D, B, A   (٢)
30 FORMAT(1X, 2A1, 1X, 2A4)

```

```

WRITE(6, 20) A, B   (١)
20 FORMAT(1X, A8, A8)

```

(أ) تخزن البيانات الحرفية ، كالآتي :

A	E	R	I	C
B	S	O	N	b
C	M	O	R	T
D	Y	b	b	b

(ب) (!) يهدف إشارة التحكم في الرتبة $1 \times$ ، تكون عروض الحقول لـ A و B هما 8 و 8 . وتطبع المحتويات من المخزن مفسطة من جهة اليمين في الحقول الخاصة بها كما يلي :

bbbbERICbbbbSONb

لاحظ أن $w - M = 8 - 4 = 4$ وهي عدد المسافات التي تسبق ERIC و SONb .

(ii) عروض الحقول A و B و C و D هي 4 و 4 و 1 و 1 على الترتيب . تبتر السلاسل الحرفية في C و D من اليمين ، ومن ثم ، يطبع :

MYbSONbERIC

٩ - ٤ افترض أن $M = 4$ وأن بطاقتي البيانات الأوليين ثقتنا كالتالي :

	1	2	3	4
البطاقة الأولى	J E F F E R S O N	T H O M A S		
البطاقة الثانية	W A S H I N G T O N	G E O R G E		

أوجد المخرج لكل من أجزاء البرامج التالية :

```

DIMENSION NAME(10) (ب)
DO 100 I = 1, 2
  READ(5, 35) NAME
  FORMAT(5X, 10A4)
  WRITE(6, 45)
  FORMAT(5X, 10A3)
100 CONTINUE
DO 100 I = 1, 2
  READ(5, 25)
  FORMAT(5X, 30HAAAAAAAAAABBBBBBBBBBCCCCCCCC)
  WRITE(6, 25)
100 CONTINUE
DIMENSION NAME(30) (أ)
DO 100 I = 1, 2
  READ(5, 60) NAME
  FORMAT(5X, 30A1)
  WRITE(6, 60) NAME
100 CONTINUE

```

(أ) يوزن كل حرف في عنصر من المجموعة المترابطة NAME ثم يطبع الحرف بعد ذلك . من ثم ، يظهر المخرج كما في بطاقة البيانات :


```
JEFFERSON    THOMAS
WASHINGTON   GEORGE
```

(ب) لاحظ أنه تم تخصيص أربعة حروف لكل عنصر من المجموعة المتراسة NAME ولكن ثلاثة حروف فقط ، أي الثلاثة الأولى ، الحروف هي التي تطبع . ومن ثم سيظهر المخرج كما يلي :

```
JEFERSN      HOMS
WASINGON     EORE
```

(ج) تحمل الـ 30 حرف في الأعمدة 6 — 35 على الحروف التالية لـ 30 H في جملة FORMAT وبعد ذلك تطبع الحروف . من ثم يظهر المخرج كما هو في بطاقة البيانات :

```
JEFFERSON    THOMAS
WASHINGTON   GEORGE
```

٩ - ٥ افترض أن جملة كاملة S قد ثبتت على بطاقة واحدة ابتداء من عمود I . احسب عدد الكلمات N في S .

بفرض أن الكلمات تنفصل عن بعضها بواسطة مسافة واحدة وأن الجملة تنتهي بنقطة ، فإن N تساوى عدد المسافات التي تسبق النقطة مضافاً إليها واحد . وحيث أن البطاقة تحتوي 80 عموداً ، فنحن نجلتينا في مجموعة متراسة خطية STRING بها 80 عنصر . وفيما يلي البرنامج :

```
INTEGER STRING(80), BLANK, PERIOD
DATA BLANK/' ', PERIOD/'.'/
READ(5, 10) STRING
10 FORMAT(80A1)
N = 0
DO 100 K = 1, 80
C      TEST IF IT IS A BLANK CHARACTER
      IF(STRING(K).EQ.BLANK) N = N + 1
C      TEST IF IT IS THE END OF THE SENTENCE
      IF(STRING(K).EQ.PERIOD) GO TO 55
100 CONTINUE
55 N = N + 1
WRITE(6, 20) N
20 FORMAT(1X, 'THE NUMBER OF WORDS = ', I3)
STOP
END
```

٩ - ٦ تم تثقيب قائمة لا تتجاوز 500 اسم على بطاقات ، كل اسم على بطاقة ، تم تثقيب الأسماء الأخيرة في الأعمدة من 1 إلى 15 وتم تثقيب الأسماء الأولى في الأعمدة من 17 إلى 32 وأضيفت بطاقة خلية بها XXX مثقبة في الأعمدة من 1 إلى 3 . بفرض أن $M = 4$ اكتب جزء برنامج فورتران ليحسب عدد الأسماء N ويخزنها في مجموعة متراسة .

حيث أن السمة $M = 4$ فيحتاج كل اسم إلى عدد $8 = 32/4$ من أماكن التخزين على الأقل وذلك لتخزين الحروف 32 الممكنة ومن ثم ستخزن الأسماء في NAME مجموعة متراسة 8×500 اسم واحد في كل صف من المجموعة المتراسة . وعلاوة على ذلك ، كل مرة نقرأ بطاقة ، نتحقق من ظهور XXX في أول مكان تخزين . وفيما يلي البرنامج :

```

DIMENSION NAME(500, 8)
DATA JJ/'XXX'/
N = 0
DO 100 K = 1, 500
    READ(5, 10) (NAME(K, L), L = 1, 8)
10    FORMAT(8A4)
    IF(NAME(K, 1).EQ.JJ) GO TO 88
    N = N + 1
100 CONTINUE
88 .....

```

٩ - ٧ افترض أن $M=4$ و $NAME$ مجموعة مترابطة (IXJ) تحتوى إسماً واحداً في كل صف اكتب برنامج فرعي SUBROUTINE :

```
ALPHA(NAME, I, J, K, L)
```

يبدل الأسماء رقم K و L إن لم يكونا في الترتيب الأبجدي .

تفرز الأسماء أبجدياً كما لو كنا نفرز مكائهم الرقي ، فبا عدا الإسمين الأولين إذا كان لها نفس الحروف الأربعة الأولى ، أى ،

```
NAME(K, 1) = NAME(L, 1)
```

إذن يجب أن نقارن الحروف الأربعة الثانية ، وهكذا . وفيما يلي البرنامج :

```

SUBROUTINE ALPHA(NAME, I, J, K, L)
DIMENSION NAME(I, J)
DO 99 N = 1, J
    IF(NAME(K, N).GT.NAME(L, N)) GO TO 50
    IF(NAME(K, N).LT.NAME(L, N)) GO TO 40
99 CONTINUE
40 RETURN.
50 DO 88 M = N, J
    JSAVE = NAME(K, M)
    NAME(K, M) = NAME(L, M)
    NAME(L, M) = JSAVE
88 CONTINUE
RETURN
END

```

لاحظ أن تأثير الجملة المرقمة 50 يكافئ :

```
50 DO 88 M = 1, J
```

وسوف ندع القارئ يعطى السبب لذلك .

٩ - ٨ افترض أن عدد N من الأسماء مخزن في الذاكرة في مجموعة مترابطة $NAME$ (500×8) إسم واحد في كل صف . اكتب جزء برنامج لفرز المجموعة المترابطة $NAME$ أبجدياً .

افترض A_1 و A_2 و ... و A_N هي الأسماء . سنستخدم النظام الحسابي (الخوارزم) للفرز الفقاعي الذي تمت مناقشته في مسألة ٦ - ٢ و رسم ٨ - ٢ ، فبا عدا أننا نرتب الأسماء أبجدياً فضلاً عن ترتيب الأرقام . باستخدام البرنامج

الفرعي ALPHA في مسألة ٩ - ٧ ، نفرز A_1 و A_2 أبجدياً ثم A_3 و A_2 وهكذا إلى A_{N-1} و A_N . بعد هذا المرور خلال العناصر ، يصبح الإسم الصحيح في المركز الأخير . نكرر هذه العملية مع الأسماء A_1 و A_2 و ... و A_{N-1} بعد المرور خلال الأسماء ، يصبح الإسم الصحيح في المركز قبل الأخير ونستمر في ذلك وبعد عدد $N - 1$ من اللغات ، ستكون الأسماء في الترتيب الأبجدي .

لترجمة النظام الحسابي (الخوارزم) إلى الفورتران ، نستخدم حلقات DO المتداخلة ، يتنبر الدليل الخارجي I من 1 إلى $N - 1$ والدليل الداخلي J من 1 إلى $N - I$ وفيما يلي البرنامج :

```

NN = N - 1
DO 99 I = 1, NN
    JJ = N - I
    DO 88 J = 1, JJ
        CALL ALPHA(NAME, 500, 8, J, J + 1)
88     CONTINUE
99     CONTINUE

```

تفسر أنت منطقية وموصلات

٩ - ٩ افرض أن بطاقة بيانات ثقت كما يلي :

```

      1           2           3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
NOW IS THE TIME TO RETURN

```

يفرض أن أى حرف غير خال في الحقل غير T سيقبل على أنه . FALSE . (أنظر ص ٢٦٤) ، أوجد قيم I و J و K و X و Y إذا كان الأمر هو :

```

LOGICAL I, J, K, X, Y
READ(5, 10) I, J, K, X, Y
10  FORMAT(L6, L3, 3L6)

```

عروض الحقول هي 6 و 3 و 6 و 6 و 6 على الترتيب . أول حرف غير خال في الأعمدة من 1 إلى 6 هو N وفي الأعمدة من 7 إلى 9 هو T في الأعمدة من 10 إلى 15 هو E في الأعمدة من 16 إلى 20 هو T وفي الأعمدة من 22 إلى 27 هو T . من ثم ، نقرأ . FALSE . في I و K ونقرأ . TRUE . في J و X و Y .

٩ - ١٠ اكتب مايلي بالفورتران :

- توقف إذا كانت $J = K$ و $N > 15$.
- إذهب إلى الجملة المرقمة 71 . إذا كانت $X \geq Y$ أو $A + B < C$.
- إذهب إلى الجملة المرقمة 72 إذا كانت A ليست أكبر من B .
- توقف أن لم تكن $A < B$ و $C \geq D$.

تذكر أن "and" و "or" و "not" يرمز إليها في الفورتران بـ : «AND.» ، «OR.» ، و «NOT.» . على الترتيب .

IF(.NOT.A.GT.B) GO TO 72 (ج) IF(J.EQ.K.AND.N.GT.15) STOP (أ)
IF(.NOT.(A.LT.B.AND.C.GE.D)) STOP (د) IF(X.GE.Y**2.OR.(A + B).LT.C) GO TO 71 (ب)

٩ - ١١ افترض أن K تحتوي على 5 وأن A و B متغيرات منطقية . أوجد القيمة النهائية لـ A بعد كل جزء برنامج فورتران مما يلي :

(أ) A = .FALSE. (ب) B = .TRUE. (ج) B = 10.LT.3*K
B = 2*K.LT.15 A = K**2.GT.4*K
A = A.AND.B A = A.AND.B IF(B) A = .NOT.A

(أ) لا تتحقق A بواسطة السطر الأول وتتحقق B بواسطة السطر الثاني ، من ثم ، تتحقق قيمة A.OR.B وتصبح هي القيمة النهائية لـ A .

(ب) تتحقق B بواسطة السطر الأول ، ولا تتحقق A بواسطة السطر الثاني ، من ثم ، لا تتحقق قيمة A.AND.B وتصبح هي القيمة النهائية لـ A .

(ج) تتحقق B بواسطة السطر الأول ، وتتحقق A بواسطة السطر الثاني وحيث أن B تحققت وأن A = .NOT.A ، فن ثم تصبح قيمة A لا تتحقق .

٩ - ١٢ افترض أن A و B و C متغيرات منطقية تحتوي TRUE.. و FALSE. و TRUE. على الترتيب . أوجد المخرج لكل أمر مما يلي :

(أ) WRITE(6, 10) A, B, C
10 FORMAT(1X, 3L8)
(ب) WRITE(6, 20), A, B, C
20 FORMAT(1X, L4, 3X, L4, 3X, L4)

في المخرج يطبع كود الشكل Lw عدد 1 - w من المسافات متبوعة بـ T أو F تبعاً لما إذا كان المتغير المنطق المتناظر يتحقق أولاً يتحقق من ثم :

(أ) تطبع T و F و T في الأعمدة 8 و 16 و 24 على الترتيب .

(ب) تطبع T و F و T في الأعمدة 4 و 11 و 18 على الترتيب .

مسائل تكميلية

بيانات حرفية

٩ - ١٢ افترض أن سمة الحرف M = 4 أوجد البيانات في المخزن بعد تنفيذ ما يلي :

DATA I/'YES', J/'GO NOW'/

٩ - ١٤ افترض أن M = 4 و تم تثقيب بطاقة بيانات كالتالي :

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
LENA										AUDREY										ERIC				

أوجد البيانات في المخزن إذا كان أمر الإدخال هو :

```

DIMENSION K(5) (أ)
READ(5, 30) K
30 FORMAT(5A4)
DIMENSION L(4) (د)
READ(5, 40) L
40 FORMAT(4A5)
READ(5, 10) J (ب)
10 FORMAT(A2)
READ(5, 20) J (ب)
20 FORMAT(A7)

```

٩ - ١٥ افترض أن $M = 4$ وتم تخزين LONDON ، في IA و IB كالتالي :

IA

L	O	N	D
---	---	---	---

 IB

O	N	b	b
---	---	---	---

أوجد المخرج إذا كانت جملة WRITE هي :

WRITE(6, 55) IA, IB

وجملة FORMAT المصاحبة هي :

```

55 FORMAT(1X, A2, A3) (أ)
55 FORMAT(1X, A3, A2) (د)
55 FORMAT(1X, 2A4) (ب)
55 FORMAT(1X, 2A7) (ج)

```

٩ - ١٦ افترض أن $M = 4$ وتم ترتيب أول بطاقتي بيانات كالتالي :

	1					2					3														
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
بطاقة 1	BROWN										ROBERT														
بطاقة 2	JOHNSON										WILLIAM														

أوجد المخرج لكل جزء برنامج ما يلي :

```

DIMENSION NAME(6) (أ)
DO 99 K = 1, 2
  READ(5, 15) NAME
  15 FORMAT(10X, 6A4)
  WRITE(6, 15) NAME
99 CONTINUE

DIMENSION NLAST(3), NFIRST(3) (ب)
DO 99 K = 1, 2
  READ(5, 15) NLAST, NFIRST
  15 FORMAT(10X, 6A4)
  WRITE(6, 15) NLAST, NFIRST
99 CONTINUE

```

```

DO 99 K = 1, 2                                (ب)
  READ(5, 15)
15  FORMAT(10X, 24HAAAAAAAAABBBBBBBBBCCCCCCCC)
  WRITE(6, 15)
99  CONTINUE

DIMENSION NAME(6)                             (د)
DO 99 K = 1, 2
  READ(5, 15) NAME
15  FORMAT(10X, 6A4)
  WRITE(5, 25) NAME
25  FORMAT(10X, 6A2)
99  CONTINUE
    
```

٩ - ١٧ نفذ مسألة ٩ - ٥ ، عدا أن كلمات الجملة الآن مفصولة بمسافة أو أكثر .

٩ - ١٨ افترض أنك أعطيت مجموعة بطاقات لا تتجاوز 80 بطاقة . تحتوي كل بطاقة على اسم طالب في الأعمدة من 1 إلى 32 وحرف الدرجة في عمود 34 . حروف الدرجات المقبولة هي A و B و C و D و F و I (لمن لم يستكمل المنهج) افترض أن بطاقة خلفية قد أضيفت ومثقب عليها XXX في الأعمدة من 1 إلى 3 اكتب برنامج :

(أ) يحسب عدد الطلبة الحاصلين على كل درجة .

(ب) يطبع أسماء الطلبة في كل مستوى درجة .

٩ - ١٩ افترض أن قصيدة P تم تثقيبها في مجموعة من البطاقات حيث يثقب على كل بطاقة سطر من القصيدة ، وافترض أن المجموعة لها بطاقة خلفية مثقب عليها XXX في الأعمدة من 1 إلى 3 . (افترض أن القصيدة P ليس بها أكثر من 20 سطراً) اكتب البرنامج الذي يطبع القصيدة P وعدد البطاقات في المجموعة ، أي ، عدد الأسطر في P .

٩ - ٢٠ للقصيدة P في المسألة ٩ - ١٩ ، اكتب البرنامج الذي يحسب عدد المرات التي ظهرت فيها الحروف A و E و I و O و U في القصيدة P .

٩ - ٢١ في مسألة ٩ - ١٩ ، اكتب البرنامج الذي يحسب عدد الأسطر التي تبدأ بكلمة THE في القصيدة P (افترض أن كل سطر في القصيدة يبدأ في عمود 1 من قائمة البيانات) .

٩ - ٢٢ في مسألة ٩ - ١٩ ، اكتب البرنامج الذي يحسب عدد المرات التي تظهر فيها كلمة THE في القصيدة P (افترض أن كلمة THE تكون متبوعة دائماً بمسافة خالية ومسبوقه دائماً بمسافة خالية فيما عدا عند بداية السطر) .

٩ - ٢٣ افترض أنه تم تثقيب قصة قصيرة SS على مجموعة بطاقات ، وافترض أن المجموعة لها بطاقة خلفية مثقب عليها XXX في الأعمدة من 1 إلى 3 . اكتب البرنامج الذي يطبع القصة القصيرة SS وعدد الأسطر في SS . (افترض أن القصة ليس بها أكثر من 400 سطراً باستبعاد العنوان المثقب في أول بطاقة واسم الكاتب المثقب في ثاني بطاقة) .

٩ - ٢٤ في مسألة ٩ - ٢٣ ، اكتب البرنامج الذي يحسب عدد الفقرات في القصة القصيرة SS (افترض أن كل فقرة مثقبة على بطاقة جديدة من عمود 6 وغير ذلك كل البطاقات مثقبة لإبتداء من عمود 1) .

٩ - ٢٥ افترض عنوان به أقل من 80 حرفاً مثقب على بطاقة لإبتداء من عمود 1 . اكتب البرنامج الذي يطبع العنوان بحيث يكون متوسطاً في الأعمدة من 1 إلى 80 .

٩ - ٢٦ إفرض قائمة N من الأسماء ($N \leq 500$) خزنت في مجموعة متراسة (500×8) باسم NAME إسم واحد لكل صف . إفرض مجموعة متراسة خطية صحيحة PERSON(1) و ... و PERSON(8) تحتوي على إسم آخر اكتب جزء البرنامج الذي يحدد ما إذا كان PERSON موجوداً في NAME . إذا كانت الإجابة نعم ، أوجد المكان L للاسم ، وضع $L = 0$ فيما عدا ذلك .

٩ - ٢٧ إفرض قائمة N من الأسماء ($N < 500$) خزنت في مجموعة متراسة (500×8) باسم NAME والمراد إضافة إسم مخزن في المجموعة المتراسة الصحيحة PERSON(1) و ... و PERSON(8) في الصف L من المجموعة المتراسة NAME وذلك بتحريك الأسماء في الصفوف L و L + 1 و ... و N إلى أسفل مكان واحد . أكتب جزء برنامج لإنجاز هذه المهمة .

٩ - ٢٨ إفرض قائمة N من الأسماء ($N \leq 500$) خزنت في مجموعة متراسة (500×8) باسم NAME مرتبة ترتيباً أبجدياً . استخدم طريقة البحث الثنائي لتحديد ما إذا كان الاسم المخزن في المجموعة المتراسة الصحيحة PERSON(1) و ... و PERSON(8) موجوداً في المجموعة المتراسة NAME . إذا كان كذلك ، اجعل المتغير المنطقي IN به القيمة TRUE . وأوجد مكان الصف L . إن لم يكن كذلك ، اجعل IN به القيمة FALSE . وأوجد مكان الصف L حيث يجب أن يضاف هذا الإسم ، ثم أضف الإسم كما في المسألة ٩ - ٢٧ .

متغيرات منطقية وروابط

٩ - ٢٩ إفرس المتغيرات المنطقية الوحيدة في البرنامج هي A و B و C و D اكتشف الأخطاء ، إن وجدت ، في كل جملة :

$$\begin{array}{lll} F = C.AND.D & (أ) & C = R.GT.(A + 2.0) & (ب) & A = FALSE & (أ) \\ A = A.AND.TRUE. & (و) & D = A.OR.B & (د) & B = Y.GT.X & (ب) \end{array}$$

٩ - ٣٠ إفرض أنه تم تثقيب بطاقة بيانات كما يلي :

```

      1           2
     1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
    _____
TOM LEFT. IN THE NIGHT

```

يفرض أن المترجم يعتبر أول حرف غير T لا يتحقق . أوجد الخرج لكل جزء برنامج بما يلي :

LOGICAL X, Y, Z	(ب)	LOGICAL A(6)	(أ)
READ(5, 15) X, Y, Z		READ(5, 15) A	
15 FORMAT(3X, 3L4)		15 FORMAT(5L4)	
WRITE(6, 25) X, Y, Z		WRITE(6, 25) A	
25 FORMAT(1X, 3L4)		25 FORMAT(1X, 5L4)	

٩ - ٣١ اكتب الآتي بالفورتران .

(أ) توقف إذا كانت $A > B$ و $C \leq 10$.

(ب) اذهب إلى الجملة رقم 100 إذا كانت A موجبة أو $L \neq J$.

(ج) اذهب إلى الجملة رقم 200 إذا كانت $7 < X < 10$.

(د) توقف أن لم يتحقق الشرط $A \leq B$ و $X \geq Y$.

٩ - ٢٢ إترض أن A و B و C متغيرات منطقية ولها التسم FALSE . و FALSE . و TRUE . على الترتيب . أوجد قيمة كل تعبير منطقي مما يأتي :

(أ) A.OR.C
(ب) B.AND.C
(ج) A.OR.NOT.C
(د) NOT.A.AND.B
(هـ) NOT.(A.AND.B)
(و) A.AND.B.OR..NOT.C
(ز) A.OR.B.AND.C
(ح) .NOT.(NOT.B.OR.A)
(ط) A.OR.C.AND.B.OR.C

٩ - ٢٢ إترض أن J تحتوي على 10 وأن A و B متغيرات منطقية . أوجد القيمة النهائية لـ A بعد كل جزء برنامج مما يلي :

(أ) A = J + 10.EQ.2*J
B = .TRUE.
IF(A.OR..NOT.B) J = J + 10
A = J.GT.15
(ب) A = .TRUE.
B = J.LT.5
A = A.AND.B
(ج) A = .TRUE.
B = .NOT.A
IF(NOT.A) B = .FALSE.
A = A.OR.B
(د) A = J.GT.15
B = .TRUE.
A = .FALSE.
IF(NOT.A) B = .FALSE.
A = A.OR.B

٩ - ٢٤ إترض أن كل من J و K و L تحتوي على 10 و 20 و 30 على الترتيب . أوجد قيمة كل تعبير منطقي .

(أ) 2*J.EQ.K.AND.K.LE.L
(ب) NOT.J.GT.K.OR.L.LT.5
(ج) K.EQ.10.AND..NOT.J.LT.L - 15
(د) NOT.(5.EQ.J - 5.AND.2*K.EQ.J + L)

٩ - ٢٥ اكتب تعبيراً منطقياً بالفورتران لكل شرط :

(أ) A أقل من كل من X و Y و Z .

(ب) تقل A عن X و Y و Z على الأقل بواحد .

(ج) $X > 2$ ولكن $Y \leq 3$.

(د) كل من A و B موجب ، أو أحدهما أكبر من 100 .

(هـ) $Y \geq Z$ و $0 < X < 1$.

اجابات للمسائل التكميلية المختارة

١٢ - ٩

YES_b في I و GO_bN في J

١٤ - ٩ (أ) LE_{bb} ، في J (ب) A_{bb}A في J (ج) LENA في K(1) ، AU_{bb} في K(2) ، DERY في K(3) ، ER_{bb} في K(4) ، IC_{bb} في K(5) ، ENA_b في L(1) ، AUDR في L(2) ، Y_{bb}E في L(3) و IC_{bb} في L(4)

١٥ - ٩

LONDON_{bb} (أ) ، ON_{bb} (ب) ، LOND_{bb} (ج) ، LOON_b (د) ، LONON (هـ)

١٦ - ٩ (أ) ، (ب) ، (ج) يظهر الحرج كما في بطاقات البيانات .

(د) BRN_{bbb}RORT_{bb} و JOSO_{bb}WIIA_{bb}

٢٩ - ٩ (أ) A = .FALSE. (ب) لا توجد أخطاء . (ج) A + 2.0 غير مقبولة حيث أن A متغيراً منطقياً .
(د) غير مقبولة حيث أن F ليست متغيراً منطقياً . (هـ) لا توجد أخطاء . (و) لا توجد أخطاء .

٣٠ - ٩

(ب) T, T, F

(أ) T, F, T, F, F, T

IF(A.GT.B.AND.C.LE.10.0) STOP (أ)

٣١ - ٩

IF(J.NE.L.OR.A.GT.0.0) GO TO 100 (ب)

IF(7.0.LT.X.AND.X.LT.10.0) GO TO 200 (ج)

IF(NOT.(A.LE.B.AND.X.GE.Y)) STOP (د)

٣٢ - ٩

T (أ) ، F (ب) ، F (ج) ، F (د) ، T (هـ) ، F (و) ، F (ز) ، F (ح) ، T (ط)

٣٣ - ٩

T (د)

T (ج)

F (ب)

F (أ)

٣٤ - ٩

F (د)

F (ج)

T (ب)

T (أ)

A.LT.X.AND.A.LT.Y.AND.A.LT.Z

(أ) ٣٥ - ٩

A.LT.X.OR.A.LT.Y.OR.A.LT.Z

(ب)

X.GT.2.0.AND.Y.LE.3.0

(ج)

A.GT.0.0.AND.B.GT.0.0.OR.(A.GT.100.0.OR.B.GT.100.0)

(د)

(0.0.LT.X.AND.X.LT.1.0).AND.(Y.GE.Z)

(هـ)

الفصل العاشر

ملاحج اضافية للإدخال / الإخراج

١٠ - ١ مقدمة

هناك ملاحج أخرى للإدخال / الإخراج لم تناقش بعد . فف الفصل الثالث ، ناقشنا مواصفات حقل I— وحقل F— وحقل E— وحقل X— وفي الفصل التاسع ناقشنا مواصفات حقل A— وحقل H— وحقل L— وسنقدم في هذا الفصل عدة ملاحج جديدة ، بالإضافة إلى أننا سوف نناقش كيف نخرج (نطبع) الرسم البياني لدالة .

أولاً ، نريد أن نذكر القارئ أن أول حرف في أي (سجل) خرج لا يطبع بواسطة آلة الطباعة ولكنه يذكر فقط للتحكم في العربة . ومثل هذه المعلومة ومعلومات أخرى عن الإدخال / الإخراج يمكن أن نجدها في الفصلين ، الثالث والتاسع .

١٠ - ٢ جملة DATA

لعطين الفأيت π . ففلا يمكن أن نستخدم جملة التخصيص :

$$PI = 3.14159$$

حيث أن هذه الجملة مفيدة ، سيخصص الفأيت 3.14159 إلى PI كلما تم تنفيذ الجملة . وحيث أن π لا تقدر (ثابتة) طوال البرنامج ، فنود أن نضج القيمة 3.14159 في PI مرة واحدة فقط ويمكن إنجاز ذلك بواسطة جملة DATA كما يلي :

DATA PI/3.14159/

أي أن كلمة DATA يتبعها إسم المتغير ، وبعد ذلك نحتاج لقيمة بفرطين مائتين وحيث أن جملة DATA جملة غير مفيدة ، لذا يجب أن توضع قبل أي جملة مفيدة في البرنامج .

يمكن أن نستخدم جملة DATA لإعطاء قيم ابتدائية للمتغيرات معينة ، ففلا يمكن أن نعطين قيمة ابتدائية للعداد KOUNT ، وليكن واحداً وكذلك المجموع التراكمي SUM وليكن صفراً . يمكن أن نجرى ذلك كما يلي :

DATA KOUNT/1/
DATA SUM/0.0/

يمكن أن نستخدم جملة DATA واحداً للتحديد عدة لقيم ، ففلا :

DATA KOUNT/1/, SUM/0.0/

أو كالتالي :

DATA KOUNT, SUM/1, 0.0/

الجملة الأخيرة لها الشكل العام التالي :

/ قائمة من الثوابت / قائمة من المتغيرات DATA

تفصل هذه المتغيرات بواسطة فاصلات ، وتفصل الثوابت بواسطة فاصلات ، وهناك تناظر واحد - إلى - واحد ما بين المتغيرات والثوابت . رغم أن جملة DATA لها نفس تأثير جعل التخصيص إلا أن هناك فرقا واضحا وهو أن جملة DATA هي جملة إعلانية غير منفذة ، لذا تتحدد قيم المتغيرات أثناء الترجمة . من ناحية أخرى ، جعل التخصيص جعل منفذة من ثم ، سيتم القيام بها أثناء التنفيذ .

ملاحظة : أننا نؤكد في مناقشتنا على « إعطاء قيمة إبتدائية » . رغم أننا يمكن أن نعطي قيمة إبتدائية لعداد KOUNT بواسطة :

DATA KOUNT/1/

إلا أنه يمكن تغير KOUNT أثناء التنفيذ . ولكن حيث أن جملة DATA غير منفذة ، فلا يمكن استدعاؤها أثناء التنفيذ لإعادة إعطاء قيمة إبتدائية للمتغير . وبالتحديد سيكون العداد KOUNT القيمة 1 طالما لم تخصص قيمة أخرى للعداد KOUNT أثناء التنفيذ ، ولكن بمجرد تخصيص قيمة أخرى للعداد KOUNT لا يمكن تنفيذ جملة DATA لإعطاء KOUNT القيمة الإبتدائية 1 مرة أخرى . يمكن أن نعطي قيم إبتدائية لعناصر متراسة باستخدام جملة DATA . وبالتحديد ، يمكن أن نعطي قيمة إبتدائية لمجموعة متراسة بكتابة إسما فقط ، مثلا :

DIMENSION GRADE(4)

DATA GRADE/92.0, 78.0, 43.5, 88.0/

تخزن 92.0 و 78.0 و 43.5 و 88.0 في GRADE(1) و ... و GRADE(4) على الترتيب . ونؤكد أننا إذا ذكرنا اسم المجموعة المتراسة ، سوف تعطى المجموعة المتراسة بأكلها قيا إبتدائية بالترتيب الذي خزنت به المجموعة المتراسة ، أى يتغير الدليل الأول أسرع ، الخ (أنظر قسم ٦ - ٦) . تسمح بعض المترجمات أيضاً باستخدام حلقة DO ضمنية في جملة DATA لإعطاء قيمة إبتدائية لجزء من المجموعة المتراسة مثلا الجمل الآلية :

INTEGER A(10)

DATA A(2)/5/, (A(I), I = 4, 7)/1, 2, 3, 4/

تخزن القيمة 5 في A(2) والقيم 1 و 2 و 3 و 4 في A(4) و A(5) و A(6) و A(7) على الترتيب .

يسمح الفورتران أيضاً باستخدام معالم تكرر في جملة DATA . وبالتحديد نكتب الرمز .

n*

قبل أى ثابت حيث n هي عدد صحيح موجب بدون إشارة وتدل على أن الثابت سيتكرر عدد n من المرات مثلا :

DIMENSION X(100)

DATA X/100*0.0/

تدل على أن القيمة 0.0 تتكرر 100 مرة ، ولذا تخصص 0.0 إلى كل عناصر المجموعة المتراسة X وعددها 100 .

يمكن أيضاً استخدام جملة DATA لتخزين ثوابت لسلاسل حرفية وثوابت منطقية (نقصد هنا بثابت سلسلة حرفية ، سلسلة حروف عمامة بفاصلات عليا ، مثلا "AVERAGE" و "THE END") .

لايسمح الفورتران القياسي لثوابت السلسلة الحرفية أن تظهر في جعل تخصيص . وبالتالي ، يجب أن نستخدم جملة READ لتخزين حروف بدلا من استخدام جملة DATA . وقد ناقشنا ذلك في قسم ٩ - ٢ . فلخص قواعدها كآآق :

يتغير الحد الأقصى M من الحروف الذي يمكن تخزينه في أى مكان ذاكرة (ويسمى سعة الحرف) من آلة إلى أخرى . إفرض أن السعة

M = 4 من ثم فإن :

DATA NAME/'PAT'/

تخزن سلسلة الحروف PAT مضبوطة من جهة اليسار في المكان المسمى NAME كالتالي :

NAME

P	A	T	b
---	---	---	---

مع إضافة مسافة واحدة إلى اليمين . في حالة السعة $M = 2$ فإنه يخزن في الذاكرة الحرفان اللذان على أقصى اليسار PA فقط .

NAME

P	A
---	---

وستعتبرها بعض المترجمات مثل WATFOR-WATFIV خطأ طالما تجاوزت طول سلسلة الحروف I السعة M . قد تخزن بعض المترجمات الحروف الزائدة في المكان التالي من المجموعة المتراسة إذا استخدمنا اسم مجموعة متراسة لإدخال الحروف (أنظر مسألة ١٠-٢) . يمكننا ببساطة كتابة مايل لتخزين قيم منطقية باستخدام جمل DATA :

LOGICAL A, B
DATA A, B/.TRUE., .FALSE./

و كما تمت مناقشته في قسم ٩ - ٨ .

مثال ١٠-١

(أ) إدرس جزء البرنامج التالي :

DIMENSION B(100)
LOGICAL L(5)
DATA B/40*0.0, 30*1.0, 20*2.0, 10*3.0/, L/3*.TRUE., .FALSE., .TRUE./

من ثم تخصص 0.0 إلى كل من B(1) إلى B(40) وتخصص 1.0 إلى كل من B(41) إلى B(70) وهكذا . تخصص أيضاً TRUE إلى L(1) حتى L(3) و L(5) ولكن تخصص FALSE إلى L(4) .

(ب) ستقوم الجمل التالية :

DIMENSION A(3, 2)
DATA A, B, C, D/1., 2., 3., 4., 5., 6., 3*9.9/

بمعل الآتي:

A(1, 1) = 1., A(2, 1) = 2., A(3, 1) = 3., A(1, 2) = 4.,
A(2, 2) = 5., A(3, 2) = 6. and B = 9.9, C = 9.9, D = 9.9

١٠ - ٣ حقل - T

يمكن أن تخبر الحاسب بمكان عمود الكتابة أو القراءة بنفس طريقة استخدام مفتاح tab على الآلة الكاتبة . وينفذ ذلك باستخدام حقل T والشكل العام لمواصفات الحقل هو :

T_p

حيث p هي ثابت صحيح بدون إشارة . وتحدد p مكان عمود البداية التي سوف تقرأ منها أو نطبع إليها المعلومات .

افرض ، مثلا ، أننا نريد أن نخزن في المكانين I ، X القيم المنقبة في الأعمدة من 13 إلى 15 وفي الأعمدة من 21 إلى 28 على الترتيب فيمكن أن ننجز ذلك بالتالي :

READ(5, 80) I, X
80 FORMAT(12X, I3, 5X, F8.2)

يمكن باستخدام حقل T— أن نستخدم FORMAT المكافئة الآتية :

80 FORMAT(T13, I3, T21, F8.2)

لاحظ أن حقل T— يحدد مكان البداية لمواصفات الحقل التالية .

أحياناً ، نجد أن استخدام حقل T— أبسط حيث أنه يجنبنا عد أماكن الأعمدة .

وحيث أن كل مواصفات الحقول تشير إلى حقول متصلة بزواج الجمل READ-FORMAT .

READ(5, 22) A, B

22 FORMAT(T20, F8.2, 12X, F8.2)

تدل ضمناً أن قيمة A ستوجد في الأعمدة من 20 إلى 27 وقيمة B ستكون في الأعمدة من 40 إلى 47 .

يمكن قراءة القيم باستخدام حقل T— بأي ترتيب . فمثلا :

READ(5, 23) X, Y

23 FORMAT(T31, F8.2, T11, F8.2)

ستخصص إلى X العدد الموجود في الأعمدة من 31 إلى 38 وتخصص إلى Y العدد الموجود في الأعمدة من 11 إلى 18

بل يمكننا أيضاً أن نقرأ عنصر عدة مرات . فمثلا :

READ(5, 24) M, N, N1, N2, N3

24 FORMAT(T21, A3, T21, I3, T21, 3I1)

ستخزن المعلومات الموجودة في الأعمدة من 21 إلى 23 كسلسلة حرفية (أى ، في الشكل الأبجدي الرقى) في المكان M وقيمة صحيحة (أى في الشكل الرقى) في المكان N والثلاثيات في الأماكن N₁ و N₂ و N₃ على الترتيب .

في الخرج ، يعطى حقل T— مكان البداية في سجل الإخراج (صف) وليس على صفحة الطباعة . وحيث أن الحرف الأول في سجل

الخرج يستخدم للتحكم في العربة فسيكون مكان البداية على صفحة الطباعة أقل بواحد من الرقم الذي يظهر في مواصفات حقل T— فمثلا

WRITE(6, 28) I, J, K

28 FORMAT(15X, I7, T28, I5, T42, I9)

تأمر الحاسب أن يطبع I مضبوطة من اليمين في الأعمدة من 14 إلى 20 وأن يطبع J مضبوطة من اليمين في الأعمدة من 27 إلى 31 ويطبع K مضبوطة من اليمين في الأعمدة من 41 إلى 49 وهذه مكافئة لما يلي :

WRITE(6, 28) I, J, K

28 FORMAT(T15, I7, 6X, I5, 9X, I9)

ومن الطبيعي أنه يمكن أن نطبع القيم بأي ترتيب . فمثلا :

WRITE(6, 39) X, Y

39 FORMAT(T31, F8.2, T11, F8.2)

ستطبع Y في الأعمدة من 10 إلى 17 و X في الأعمدة من 30 إلى 37 .

١٠ - حقل G

يمكن أن يستخدم الشكل المعمم لحقل G— لإدخال / وإخراج أما بيانات صحيحة أو حقيقية أو منطقية ، أو مركبة (سوف نناقش

البيانات المركبة في الفصل الحادى عشر) الشكل العام لمواصفات الحقل هو :

Gw.d

حيث تشير w إلى عرض الحقل و d ثابت صحيح بدون إشارة . (يمكن أن نستخدم أيضاً $rGw.d$ حيث تشير r إلى عدد مرات التكرار)
يتحدد معنى $Gw.d$ بواسطة نوع المتغير المناظر كالاتي :

متغير صحيح

$Gw.d$ لها نفس معنى Iw في كل من المدخل والمخرج .

متغير منطوق :

$Gw.d$ لها نفس معنى Lw في كل من المدخل والمخرج .

متغير حقيقي

هناك حالتان :

١ - الإدخال $Gw.d$ لها نفس معنى $Ew.d$ أو $Fw.d$ تبعاً ما إذا كانت القيمة بها E مثبتة في الحقل الخاص بها أم لا .

٢ - المخرج سترمز d في هذه الحالة دائماً إلى عدد الخانات المنوية وسيكون المخرج في شكل F — إذا أمكن طباعة القيمة في حقل
بمرض $w-4$. في هذه الحالة ، ستطبع القيمة مضافاً إليها أربع مسافات فقط على الطرف الأيمن للحقل (أى ، مضبوطة من
اليمن إلى خامس عمود على يمين الحقل) . سيكون المخرج في شكل E — إذا لم يمكن طباعة القيمة في حقل بمرض $w-4$.
في هذه الحالة ، ستطبع القيمة مضبوطة من اليمين في الحقل الخاص بها .

متغير مركب

ستتم مناقشته في الفصل الحادي عشر .

مثال ١٠ - ٢

(أ) افترض أن I و J و A و B تخزن باستخدام ما يلي .

```
LOGICAL J
READ(5, 30) I, J, A, B
30 FORMAT(3G10.3, G12.3)
```

حيث تتقب بطاقة البيانات كالتالي :

```

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
-----
1 2 3   THE  END   4 . 5 6 7 8 9   - 1 2 . 3 4 5 6 E - 0 6
```

وحيث أن $G10.3$ لها معامل التكرار 3 فتكون مواصفات الحقول $G10.3$ و $G10.3$ و $G10.3$ و $G12.3$ بمرض
10 ، 10 و 12 على الترتيب .

(١) تناظر أول $G10.3$ المتغير الصحيح I لذا فسوف تفسر مثل $I10$ من ثم ، تخصص 123 إلى المتغير I وهو الرقم الصحيح
المثبت في الحقل الأول في الأعمدة من 1 إلى 10 .

(٢) تناظر ثان $G10.3$ المتغير المنطوق J ولذا تفسر مثل $L10$ وحيث أن أول حرف غير المسافات الخالية في الحقل الثاني
وفي الأعمدة من 11 إلى 20 هو T فسوف يتم تخزين $TRUE$. في J .

- (٣) تناظر ثالث G10.3 المتغير الحقيقي A : وحيث لا يوجد E مثقبة في الحقل ، أى في الأعمدة من 12 إلى 30 لذا فلها نفس معنى F10.3 ، من ثم يخصص إلى A القيمة 4.56789 .
- (٤) تناظر G12.3 المتغير الحقيقي B وحيث أن هناك E مثقبة في الحقل أى في الأعمدة من 31 إلى 42 لذا فتصبح E'2.3 . من ثم ، يخزن 06 — 12.3456E — أو المكافئ لما 0.0000123456 — في B
- (ب) افترض أن أمر الإخراج هو :

```
DO 100 K = 1, 6
    WRITE(6, 30) A(K)
30    FORMAT(6X, G12.4)
100 CONTINUE
```

حيث A مجموعة متراصة خطية بها ستة عناصر تحتوى ما يلي على الترتيب :

77.777, 0.333×10^{-5} , 2.2222×10^2 , 666.66×10^7 , 12.3, 8.88×10^{-3}

الكود G12.4 يقرب كل رقم إلى أربع خانات مئوية :

77.78, 0.00000333, 222.2, 6667000000., 12.30, 0.00888

حيث أن عرض حقل الخرج (الأعمدة من 6 إلى 17) هو 12. فتطبع الأرقام التى لا يمكن طباعتها بأربع مسافات خالية على اليمين ، أى الأرقام التى تتطلب أكثر من 4 = 3 - 4 أعمدة ، فى الشكل الاسى - وبذلك ، تطبع A(2) و A(4) فى الشكل الاسى ، ويظهر الخرج كما فى الشكل ١٠ - ١ . لاحظ أننا نستطيع أن نتعرف على الفور على الأرقام فى الشكل الاسى .



شكل ١٠ - ١

١٠ - ٥ معامل التدرج

عند التعامل مع مجموعات ضخمة من البيانات ، أحيانا يكون من المفيد أن نضمن فى حقول E أو F معامل تدرج ومعامل التدرج له الشكل :

sP

حيث تسمى s (معامل التدرج) وهى ثابت صحيح بدون إشارة أو ثابت صحيح سالب . تكتب على يسار الحقل الأول الذى سوف تطبق عليه ، فثلا :

3PF8.2 or -4PE14.4

ومع ذلك يختلف ، تأثير معامل التدرج على حقل F عن تأثيره على حقل E .

(أ) استخدام معامل التدرج مع حقل F—

عند استخدام معامل التدرج مع حقل F— يكون تأثير SP كما يلي :

$$\text{القيمة الخارجية} = \text{القيمة الداخلية} \times 10^5$$

أو

$$\text{القيمة الداخلية} = \text{القيمة الخارجية} \times 10^{-5}$$

هذا صحيح في كلا من الإدخال / الإخراج .

مثال ١٠ - ٣

(أ) افترض أن X تحتوي على الرقم 0.325 ونريد أن نطبع X كنسبة مئوية . يمكن أن نكتب :

```
WRITE(6, 10) X
10 FORMAT(1X, 2PF8.2, 1X, 'PERCENT')
```

نضرب القيمة الداخلية في 10^2 ولذا يكون المخرج في الصورة :

```
32.50 PERCENT
```

(ب) افترض أن الرقم المثقب في الأعمدة من 1 إلى 8 من بطاقة بيانات هو 7.50 وعندما ننفذ :

```
READ(5, 20) RATE
20 FORMAT(2PF8.2)
```

يكون الرقم المخصص إلى RATE هو 0.075 .

ويعني آخر ، عندما تستخدم SP مع حقل F— في الإدخال ، سيكون الرقم المخزن هو الرقم المثقب مضروباً في المعامل 10^{-5} . ولكن عندما نستخدم مع المخرج فيكون الرقم المطبوع هو الرقم الداخل مضروباً في المعامل 10^5 .

(ب) استخدام معامل التدرج مع حقل E—

عندما تستخدم معاملات التدرج مع حقول E— في الإدخال فإنه يتم إهمالها في المخرج ، لا يتغير حجم الرقم المطبوع ؛ ولكن يضرب الجزء الأساسي للصورة الأسية في معامل 10^5 ويخفض الأس بمقدار 5 . (تذكر أن الجزء الأساسي للصورة الأسية هو رقم ما بين 0.1 و 1.0 أو -0.1 و -1.0) .

فتلا ، افترض أن القيمة الداخلية X هي 0.0004321 ستظهر X إذا طبعت باستخدام مواصفات الحقل E12.4 ، في الشكل :

```
0.4321E - 03
```

من ناحية أخرى ، ستظهر X إذا طبعت باستخدام مواصفات الحقل 1PE12.4 ، في الشكل

```
4.321E - 04
```

(الآن تظهر قيمة X في الترميز العلمي ، أي ، كرقم ما بين 1.0 و 10.0 أو -1.0 و -10.0 — متبوعة بقوة 10) .

تعليق : بمجرد استخدام معامل التدرج في حقل E أو F لأول مرة فإنه يطبق أوتوماتيكياً على جميع حقول E ، F التالية في البرنامج إلى أن يقابل معامل آخر . لذلك ، يجب أن نستخدم معامل التدرج

```
0P (zeroP)
```

عندما نريد أن نلغي تأثير معامل تدرج سابق . فتلا ، افترض أن قيم A و B و C و D الداخلية هي 0.0111 و 0.0222 و 0.0333 و 0.0444 على الترتيب أو امر المخرج هي كما يلي :


```

WRITE(6, 10) A, B
10 FORMAT(1X, 1PE12.4, 5X, E12.4)
WRITE(6, 20) C, D
20 FORMAT(1X, E12.4, 5X, 0PE12.4)

```

فسوف يطبق معامل التدرج 1 P لطباعة A و B و C ولكن لا يطبق على D أو أي قيمة حقيقية و أي جملة WRITE تالية ، وبذلك سيظهر الإخراج في الشكل :

```

1.110E-02      2.220E-02
3.330E-02      0.4440E-01

```

١-٦ قاعدة الأقواس اليسرى

تذكر أنه إذا احتوت قائمة المتغيرات في جملة READ أو WRITE على عناصر أكثر من تلك التي في مواصفات المقبول في جملة FORMAT المصاحبة فإن جملة FORMAT تكرر عدة مرات (تقرأ بطاقة بيانات جديدة أو يطبع سطر جديد في كل مرة) إلى أن يتم الانتهاء من كل المتغيرات فإلا افترض أننا نقلنا الآتي :

```

READ(5, 10) A, B, C, D, E, F, G, H
10 FORMAT(F6.2, F8.3, F5.1)

```

فسوف نستخدم جملة FORMAT لقراءة قيم A و B و C (على أول بطاقة بيانات) وبعد ذلك تكرر جملة FORMAT لـ D و E و F (على ثاني بطاقة بيانات) ، وبعد ذلك تكرر لكل من G و H (على ثالث بطاقة بيانات) .

قد نرغب أحياناً في تكرار جزء فقط من جملة FORMAT والقاعدة هي تكرار جملة FORMAT ابتداء من القوس الأيسر إلى أقصى اليمين وحتى نهاية جملة FORMAT .
فإلا ، افترض أننا قمنا بتعديل جملة FORMAT السابقة كالتالي :

```

READ(5, 20) A, B, C, D, E, F, G, H
20 FORMAT(F6.2, (F8.3, F5.1))

```

بعد استخدام جملة FORMAT لـ A و B و C تكرر جملة FORMAT ولكن ابتداء من القوس الأيسر أقصى اليمين فقط ، أي ، ابتداء من مواصفات الحقل F8.3 من ثم ، تقرأ D و E من بطاقة البيانات الثانية و F و G من بطاقة البيانات الثالثة و H من بطاقة البيانات الرابعة . وكل مرة باستخدام المواصفات F8.3 و F5.1 .

نطبق القاعدة السابقة أيضاً على الأقواس التي تسبق بمعامل تكرار . نجد مثلاً

```

READ(5, 30) A, B, C, D, E, F, G, H
30 FORMAT(F6.2, 2(F8.3, 3X), F5.1)

```

تكافئ جملة FORMAT الخاصة بالجملة التالية :

```

30 FORMAT(F6.2, (F8.3, 3X, F8.3, 3X), F5.1)

```

بعد استخدام جملة FORMAT لـ A و B و C و D تكرر جملة FORMAT ابتداء من القوس الأيسر في أقصى اليمين ، أي ، من F8.3 إلى نهاية الجملة (وليس إلى نهاية القوس الداخلي) . وبذلك تخصص قيم إلى E و F و G من ثاني بطاقة بيانات باستخدام F8.3 و 3X و F8.3 و 3X و F5.1 وتخصص قيمة إلى H من ثالث بطاقة بيانات باستخدام F8.3 .

تطبق القاعدة السابقة أيضاً على جملة FORMAT التي تصاحب جملة WRITE .

١٠ - ٧ صيغ (FORMAT) وقت التنفيذ

تبعاً للملامح التي تمت مناقشتها حتى الآن ، فإنه بمجرد كتابة البرنامج فسوف يتم تحديد كل جمل FORMAT وبالتالي تكون ثابتة وقد يكون من الملائم جداً إذا أمكن تعديل جمل FORMAT أثناء التنفيذ لتناسب البيانات ، فبلا ، قد نحتاج أن نكتب عرض الحقل إذا كانت القيم أكبر مما توقعنا أصلاً . يناقش هذا القسم هذه المقدرة باستخدام مايسمى بجمل صيغ (FORMAT) ولت التنفيذ (ويحددها المستخدم) .

وكما يشير الإسم فجملة FORMAT التي تعطى (تحدد) أثناء التنفيذ تسمى بجمل صيغة (FORMAT) وقت التنفيذ . والفكرة العامة وراء صيغة (FORMAT) وقت التنفيذ هي تخزين (إدخال) مواصفات الحقل كسلسلة حروف في مجموعة متراسة . وبعد ذلك تستدعي جملة WRITE/READ المجموعة المتراسة إذا أردنا استخدام مواصفات الحقل في المجموعة المتراسة للإدخال / الإخراج .

إدرس زوج الجمل READ-FORMAT التالية :

```
READ(5, 10) A, B, I, J
10 FORMAT(2(F10.2, 2X), I5, 2X, I8)
```

بفرض أن سعة الحرف $M = 4$ ، نوضح كيف يمكن أن نحصل على نفس النتيجة باستخدام صيغ (FORMAT) وقت التنفيذ .
أولاً نكتب مايلي :

```
(2(F10.2, 2X), I5, 2X, I8)
```

على بطاقة بيانات ، وبعد ذلك نقرأ سلسلة الحروف (باستخدام حقل A) في مجموعة متراسة . وليكن :

```
INTEGER FORM(6)
:
READ(5, 20) FORM
20 FORMAT(6A4)
```

الحروف المخزنة في المجموعة المتراسة FORM هي كالآتي :

FORM(1)	(2	(F
FORM(2)	1	0	:	2
FORM(3)	,	2	X)
FORM(4)	,	I	5	.
FORM(5)	2	X	,	I
FORM(6)	8)		

لاستخدام صيغ (FORMAT) وقت التنفيذ هذه ، نكتب مايلي :

```
READ(5, FORM) A, B, I, J
```

لاستخدام أن اسم المجموعة المتراسة FORM (مكان تخزين مواصفات الحقل) يكتب بدلاً من رقم جملة FORMAT .

يمكن أن يكون المجموعة المتراسة المستخدمة لمواصفات الأشكال أى عدد من العناصر - طالما أنها كبيرة بدرجة كافية (ت
المواصفات المتوقعة) . وغالباً يستخدم ما يلى للرونة الكاملة :

```
INTEGER FORM(80)
READ(5, 40) FORM
40 FORMAT(80A1)
```

وهذا يلائم أى مواصفات مثبتة على بطاقة بيانات واحدة وبأى سعة M .
ونلخص ماسبق بأصدار قائمة النقاط الهامة التى يجب أن نتذكرها :

١ - يجب أن تخزن الصيغة (FORMAT) المطلوبة فى مجموعة متراسة حتى يمكن تخزينها فى مكان واحد فقط ، فمثلا
رغم أن :

```
(1X, I8)
```

يمكن تخزينها فى مكان واحد INFOR عندما تكون سعة الحروف $M = 9$ ، ولكن لزماً علينا أن نعلم أن
INFOR مجموعة متراسة .

٢ - عند تحضير بطاقة البيانات لأكواد الأشكال لا تثقب كلمة FORMAT ولكن تثقب مواصفات الحقول بداخل أقواس .

٣ - عند استخدام صيغة (FORMAT) محددة بالمستفيد ، يظهر إسم المجموعة المتراسة المحتوى على مواصفات الحقول فى جملة
WRITE/READ فمثلا ،

```
WRITE(6, INFOR) A, B, C, J, K
```

تخبر الحاسب أن مواصفات الأشكال مخزنة فى المجموعة المتراسة INFOR
بدلاً من قراءة مواصفات الأشكال فى مجموعة متراسة كما سبق ، يمكن أيضاً أن تعطى قيمة إبتدائية لصيغة (FORMAT)
وقت التنفيذ باستخدام جملة DATA :

```
INTEGER FORM(3)
DATA FORM/'(F8.', '3,I8', ')'/
```

إذا سمح المترجم بظهور ثوابت كسلسلة حرفية فى جمل التخصيص ، فيمكننا أن نستخدم :

```
INTEGER FORM(3)
FORM(1) = '(F8.'
FORM(2) = '3,I8'
FORM(3) = ')'
```

(ومن الطبع ، فن الممكن استخدام حقل هولوروث أيضاً) .

بمجرد تخزين مواصفات الصيغة فى مجموعة متراسة كما سبق ، فيمكن أيضاً أن نعدل مواصفات الصيغ وفقاً للبيانات
النتيجة . توضح هنا هذا الأسلوب الفنى بأمثلة ،
١ - افرض أنه تم تحديد الصيغة مسبقاً بواسطة مايل :

```
DIMENSION INFOR(3), X(100)
DATA INFOR/'(I3.', '2X,F', '6.2)'/, IBIG/'8.2)'/, ISMALL/'6.2)'/
```

معنى هذا أنه تم تخزين (3 I و 2X و F6.2) و INFOR . جزء البرنامج التالى :

```
DO 100 K = 1, 100
    IF(ABS(X(K)).GT.100.0) INFOR(3) = IBIG
100 CONTINUE
WRITE(6, INFOR) (N, X(N), N = 1, 100)
```

سيحقق أولاً من أن قيمة أى عنصر في المجموعة المتراسة X يتجاوز الرقم 100 . إن كان كذلك ، فتغير مواصفات الصيغة F6.2 إلى F8.2 وذلك للملاحج بمساحة أكبر تلائم البيانات . وبالتالي فإن :

```
DO 200 K = 1, 100
  IF(ABS(X(K)).GT.100.0) INFOR(3) = IBIG
  WRITE(6, INFOR) K, X(K)
  INFOR(3) = ISMALL
200 CONTINUE
```

تغير F6.2 إلى F8.2 فقط بالنسبة لعناصر معينة في المجموعة المتراسة والتي تتجاوز قيمتها الرقم 100 . تبقى F6.2 لكل العناصر الباقية .

٢ - افترض أننا نفضل حقل F لطباعة المجموعة المتراسة A وقد تم استخدام F12.4 . ومع ذلك كلما كبرت قيمة A(K) بحيث لاتناسب 12 عمود ، فإن الطبع بصورة شكل E يكون مقبولاً ، جزء البرنامج التالي يقوم بإجراء ذلك تماماً :

```
INTEGER XFORM, EFORM, FFORM
DIMENSION A(100), XFORM(4)
DATA XFORM/'1X,', 'I3,2', 'X,F1', '2.4)/, EFORM/'X,E1', FFORM/'X,F1'/
:
DO 300 K = 1, 100
  IF(ABS(A(K)).GT.1.0E6) XFORM(3) = EFORM
  WRITE(6, XFORM) K, A(K)
  XFORM(3) = FFORM
300 CONTINUE
```

لاحظ أن مواصفات الصيغة الأصلية هي (F12.4 و 2X و I3 و 1X) ولكن تم تغيير F إلى E كلما كانت A(K) كبيرة جداً ، أى ، كلما كبرت قيمتها عن 10^6 .

١٠ - الرسم البياني

إلى جانب استخدام الحاسب لحساب قوائم من القيم . فبإستطاعتنا أيضاً أن نستخدم الحاسب ليرسم القيم في شكل بياني . سيتكون الرسم البياني من خطوط من النقاط « » للمحورين وحرف "X" لكل نقطة (x, y) محسوبة . ويرسم عدد كاف من النقاط نحصل على صورة للرسم البياني . ونوضح فيما بعد هذا الأسلوب الفني .

افترض ، مثلاً أننا نريد أن نرسم الدالة :

$$y = 2x^3 - x^2 - 22x + 21$$

لكل من $4 \leq x \leq 4$ — أولاً يجب أن نختار قيم لـ x بين -4 ، 4 ، وذلك لحساب y وليكن :

$$x = -4, -3.9, -3.8, \dots, 3.9, 4$$

لاحظ أن هناك 81 قيمة لـ x وأن معامل الزيادة 0.1 . ولكننا نحتاج أيضاً مدى y لهذه القيم . افترض أنه $50 \leq y \leq 50$ — تناظر $4 \leq x \leq 4$ — بعد ذلك نجعل كل مكان على محور y يدل على القيم المتزايدة لـ y . إذا اخترنا معامل الزيادة 1.0 ، فالأماكن على محور y تمثل القيم :

$$y = -50, -49, -48, \dots, 49, 50$$

لاحظ أن هناك 101 قيمة لـ y .

هناك طريقتان لاستكمال المسألة .

الطريقة الأول :

نختار المحور الأفقي (بعرض صفحة الطباعة) كحور x والمحور الرأسى (بطول صفحة الطباعة) كحور y . فى هذه الحالة ، يجب أن نحسب كل القيم (y, x) ونخزن الرسم البيانى بأكله فى مصفوفة G قبل أن نتسكن من تنفيذ الرسم البيانى . السبب فى ذلك هو أن وحدة الطباعة تطبع سطرأ واحداً فى المرة ، ومن الممكن أن يكون لقيمتين مختلفتين لـ x نفس قيمة y (الأفقية) . (ليس فى الإمكان أن نحرك العربة أعلى وأسفل صفحة الطباعة) وبذلك سيكون لـ G صف تناظر أماكن y 101 و 81 عموداً تناظر 81 قيمة لـ x أى أن G ستكون مصفوفة (101×81) . أسوأ ما فى هذه الطريقة هو أن الرسم البيانى يحتاج مصفوفة بها $81 \times 101 = 8181$ خلية ذاكرة ولهذا السبب فعادة ترسم الرسوم البيانية باستخدام الطريقة الثانية .

الطريقة الثانية :

نختار المحور الأفقى كحور y والمحور الرأسى كحور x (يستطيع القارئ أن يتقلب الصفحة ليتابع الرسم) حيث أن كل قيمة لـ x تعطى قيمة فريدة لـ y ، نستطيع أن نحسب ونرسم رسماً البيانى سطر واحد فى المرة . بالتالى ، سنحتاج فقط لمجموعة متراسة خطية بها 101 خلية .

بالتحديد نتبع الخطوات التالية :

١ - إجعل قيمة x مساوية لقيمتها الصغرى .

٢ - أوجد القيمة المناظرة لـ y .

٣ - استخدم مقياس رسم لقيمة y رقم صحيح J يتراوح بين 1 و 101

٤ - اطبع سطرأ به حرف "X" فى العمود J من الرسم البيانى .

٥ - زد قيمة x وكرر الخطوات من ١ إلى ٤ طالما لم تتجاوز x قيمتها العظمى .

الخطوات السابقة واضحة المعالم مباشرة . وفى الخطوة الرابعة نريد أيضاً أن نطبع نقطة (٠) للعمود الذى يمثل محور x (عند $y = 0$) ونريد أيضاً أن نطبع سطرأ كاملاً من النقط (٠٠٠٠) عندما يمثل السطر محور y (عند $x = 0$) . هذه المهام مفصلة ولكنها ليست صعبة .

نحصل على دالة مقياس الرسم فى الخطوة ٣ كما يلى . من المعلوم أن y تقع ما بين 50 - و 50 يجمع 51 على y نتيج رفاً حقيقياً موجباً بين 1 و 101 . حيث نريد تقريب y إلى رقم صحيح بين 1 و 101 ، بدلاً من البتر ، يجب أيضاً أن نجمع 0.5 إلى y . يعطى هذا دالة مقياس الرسم :

$$JSCALE(Y) = INT(Y + 51.0 + 0.5)$$

لاحظ أن $JSCALE(0) = 51$ أى أن عمود 51 من الرسم البيانى هو محور x .

وقبيل البرنامج الذي يرسم دالتنا :

```

INTEGER BLANK, DOT, CROSS, LINE(101), LINEY(101)
DATA BLANK, DOT, CROSS, LINE, LINEY/' ', '.', 'X', 101*'.', 101*'. /
C
C BEGIN NEW PAGE AND SKIP LINES
C
WRITE(6, 10)
10 FORMAT('1', 10X, 'GRAPH OF A FUNCTION'//3X, 'X', 6X, 'Y'//)
C
C BEGIN GIVING VALUES TO X
C
X = -4.0
50 Y = ((2.0*X - 1.0)*X - 22.0)*X + 21.0
J = INT(Y + 51.5)
C TEST IF Y AXIS
IF(ABS(X).LT.0.001) GO TO 100
LINE(51) = DOT
LINE(J) = CROSS
WRITE(6, 20) X, Y, LINE
20 FORMAT(1X, 2(F6.2, 1X), 101A1)
LINE(J) = BLANK
GO TO 200
100 LINEY(J) = CROSS
WRITE(6, 20) X, Y, LINEY
200 X = X + 0.1
IF(X.LE.4.0) GO TO 50
STOP
END

```

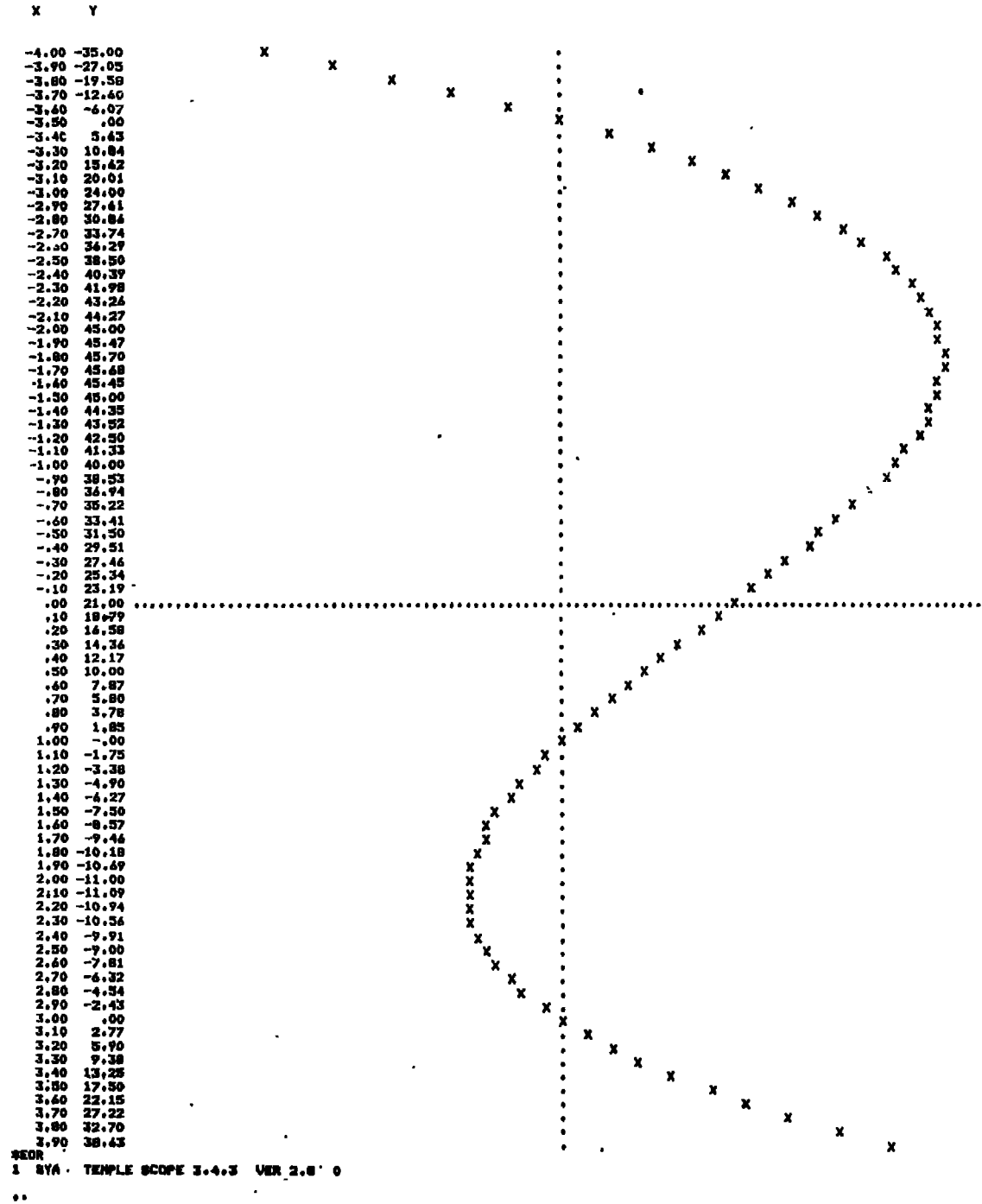
لاحظ أننا استخدمنا جملة DATA لتخزين الحروف b (مسافة) ، ، (نقطة) و X (حرف X) في المتغيرات BLANK و DOT و CROSS على الترتيب . نستخدم LINE لطباعة كل سطر من الرسم البياني إلا عندما يمثل السطر محور Y . في هذه الحالة نستخدم LINE Y من ثم ، يعطى LINE قيمة ابتدائية « مسافات » ولكن LINE Y تعطى قيمة ابتدائية « نقط » . (كان يمكننا أن نستخدم بحره متراصة واحدة ولكن حينئذ كان يجب أن نعيد مساواة المجموعة المتراسة بمسافات أو بنقط بعد كل زيادة في X) . السطر التالي :

LINE(J) = BLANK

هام فهو يمسح حرف X في LINE (J) بعد طباعته - واننا نذكر أنه يمكن أن نستخدم حلقة DO لقيم X المختلفة . يظهر خروج البرنامج في شكل ١٠ - ٢ لاحظ أن قيم X و Y لا تظهر على الجانب الأيسر من الرسم البياني .

مقدمة : تتطلب دالة مقياس الرسم JSCALE(Y) التي تمت مناقشتها فيما سبق وفي المسألة ١٠ - ١٦ ، أن نعرف مسبقاً مدى قيم Y . في حالة ما إذا لم تكن هذه المعلومات متوافرة لدينا نستطيع أن نواصل كما يلي : (١) احسب كل قيم Y و X في مجموعة متراصة Y (تبيل عمل أي رسم) . (٢) أوجد القيم العظمى والصغرى في المجموعة المتراسة Y . (٣) استخدم هذه القيم للحصول على دالة مقياس رسم JSCALE كما في المسألة ١٠ - ١٦ . (٤) ارسم الرسم البياني باستخدام الدالة JSCALE والمجموعة المتراسة Y . (تم توصيف هذا الإجراء أيضاً في المسائلين ١٠ - ٣٦ و ١٠ - ٣٧) .

1 رسم بياني لدالة



شكل ١٠ - ٢

مسائل محلولة

جملة DATA :

DATA A, B, C /2.4, '2.4', 1.5/	(١)
DATA A, B /2*2.4/, C/1.5/	(٢)
DIMENSION A(10)	(٣)
DATA A, B, C/6*1.0, 6*2.0/	(٤)
LOGICAL C	(٥)
DATA A, B, C /2.4, 'TRUE', .TRUE./ .	

(١) تخصص القيم بين الشرطات المائلة لكل من A و B و C من ثم تخصص 2.4 إلى A تخصص 1.5 إلى C وتخصص سلسلة الحروف التالية إلى B :

B	2	.	4	b
---	---	---	---	---

(مع فرض أن سعة الحرف = 4 M)

(ب) 2* هي معامل التكرار . من ثم تخصص 2.4 إلى كل من A و B وتخصص 1.5 إلى C .

(ج) 6* هي معامل التكرار . ومن ثم تخصص 1.0 إلى أول ستة عناصر من A وتخصص 2.0 إلى B ، C وإلى آخر أربعة عناصر من A .

(د) تخصص 2.4 إلى A وتخصص سلسلة الحروف .TRUE. ، إلى B وتخصص القيمة المنطقية . TRUE . إلى C .

١٠- ٢ أوجد المدخلات

LOGICAL A(3, 2)
DATA A/2*.TRUE., 3*.FALSE., .TRUE./

ترتب المجموعة المترامية A بحيث يكون الدليل الأول (دليل الصف) هو الذي يتغير أسرع . من ثم تحس .TRUE. إلى A(1, 1) و A(2, 1) وتخصص .FALSE. إلى A(3, 1) و A(1, 2) و A(2, 2) وتخصص .TRUE. إلى A(3, 2)

١٠- ٣ إذا فرضنا أن سعة الحرف = 4 M أوجد البيانات في الخزن عند تنفيذ مايلي :

DIMENSION L(2)
DATA L/'SLEEP', 'IN'/

حيث أن طول أول سلسلة حروف L تتجاوز السعة M ، فيمكن حدوث أي مما يلي :

(١) تخزن كل سلسلة حروف مضطربة من اليسار في الذاكرة مع بتر الحروف الزائدة أو تضاف مسافات زائدة على اليمين .

L(1)	S	L	E	E	L(2)	I	N	b	b
------	---	---	---	---	------	---	---	---	---

(٢) تملأ رسالة خطأ .

(٣) حيث أن إم المجموعة المترامية L يتم استخدامه في جملة DATA فإن SLEEP تخزن مضبطة من اليسار في L(1) وتحرك الحروف الزائدة إلى المكان التالي L(2) كما يلي :

L(1)

S	L	E	E
---	---	---	---

 L(2)

P	I	N	b
---	---	---	---

(نؤكد أن كل المترجمات تغطي نفس النتيجة إذا لم يتجاوز طول سلسلة الحروف ائسة M) .

حقل — T

١٠-٤ : إفرض أنه تم تثقيب بطاقة بيانات كما يلي :

1	2	3	
1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5
1 1 1 1	2 2 2 2	THE END	3 3 3 3

أوجد المدخلات إذا نفذنا الآتي :

```

READ(5, 10) I, J                                (أ)
10 FORMAT(T6, I4, T3, I6)
LOGICAL J                                        (ب)
READ(5, 20) I, J, X, Y
20 FORMAT(T11, A4, T11, L4, T21, F5.2, T3, F6.3)

```

(أ) يخصص الرقم الصحيح الموجود في الأعمدة من 6 إلى 9 إلى I أي ، الحقل الذي يبدأ في العمود 6 ويعرض 4 . من ثم فيخصص الرقم 2222 ؛ إلى I ، ويخصص الرقم الصحيح الموجود في الإعمدة من 3 إلى 8 إلى J أي في الحقل الذي يبدأ في العمود 3 ويعرض 6 . وحيث أن المسافات (الفراغات) تفسر كأصفار في الحقول الرقية ، فتخصص 110222 إلى J .

(ب) تخصص سلسلة الحروف «THE» إلى I وهي الحروف الموجودة في الحقل الذي يبدأ في العمود 11 يعرض 4 . تخصص TRUE إلى J حيث أن J متغير منطقي ، T هي أول حرف غير خال في الحقل ذي العرض 4 الذي يبدأ في العمود 11 . (لاحظ أن كلا من I و J يستخدم البيانات المثقبة في الأعمدة من 1 إلى 4) . وحيث أن T₂₁ و T₃ تأمر وحدة قراءة البطاقات بالذهاب إلى عمود 21 ثم 3 على الترتيب ، تخصص 333.30 إلى X وتخصص 110.222 إلى Y .

١٠-٥ : أوجد الخرج إذا نفذنا البرنامج الآتي :

```

I = 111
J = 222
K = 333
WRITE(5, 30) I, J, K
30 FORMAT(T15, I3, T6, I6, T20, I8)

```

تشير T_p في الخرج إلى المكان p في سجل الإخراج (-سجل) من م تشير T_p إلى العمود رقم 1 - p على صفحة الطباعة حيث لا يطبع أول حرف في السجل . بالتالي ، تطبع I مضبطة من اليمين في الحقل ذي العرض 3 الذي يبدأ في العمود 14 على صفحة الطباعة ، أي في الأعمدة من 14 إلى 16 ؛ تطبع J مضبطة من اليمين في الحقل ذي العرض 6 الذي يبدأ في العمود 5 ، أي في الأعمدة 5 إلى 10 وتطبع K مضبطة من اليمين في الحقل ذي العرض 8 الذي يبدأ في العمود 19 أي في الأعمدة من 19 إلى 26 . أنظر شكل ١٠ - ٣ .

1 2 3
1234567890123456789012345



شكل ١٠ - ٣

حقل G

١٠-٦ افرض أنه تم تثقيب بطاقة بيانات كالآتي :

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

أوجد المخرج إذا نفذنا الآتي :

```
LOGICAL Z
READ(5, 10) J, A, B, Z
10 FORMAT(4G5.1)
```

هناك 5 مداخل G5.1 ، مدخل لكل متغير . لذا فهناك أربعة حقول كل منها يرمز د .

(أ) حيث أن J متغير صحيح ، لذا تفسر G5.1 مثل I5 . ومن ثم تخصص 1110 إلى J وهو الرقم الصحيح الظاهر في الأعمدة من 1 إلى 5 (اعتبرنا المسافة الخالية في عمود 5 صفراً) .

(ب) حيث A متغير حقيقي وليس هناك E مثقبة في الأعمدة من 6 إلى 10 ومن ثم ، تفسر G5.1 مثل F5.1 . وحيث أن العلامة العشرية غير موجودة في الحقل ، لذا تضاف علامة عشرية بعد مكان واحد من الجانب الأيمن للحقل ، وبذلك تخصص 222.0 إلى A .

(ج) حيث B متغير حقيقي وحيث أن هناك E مثقبة في الأعمدة من 11 إلى 15 من ثم تفسر G5.1 مثل E5.1 وحيث أن العلامة العشرية غير موجودة في الحقل ، لذا تضاف علامة عشرية بعد مكان واحد من الجانب الأيسر للحرف B وبذلك تخصص 3.3E3 أو مكائنها 3300 إلى B .

(د) حيث Z متغير منطقي لذا تفسر G5.1 مثل L5 . وحيث أن أول حرف غير خال في الأعمدة من 16 إلى 20 هو T . بذلك تخصص TRUE. إلى Z .

١٠-٧ افرض أن J تحتوي على 2345 وأن K متغير منطقي يحتوي على FALSE . أوجد المخرج إذا نفذنا الآتي :

```
WRITE(6, 20) J, K
20 FORMAT(1X, 2G10.3)
```

باستناد أمر التحكم في العربية 1X ، فمعرض الحقول J و K هي عشرة حروف لكل منهما . وحيث أن J متغير صحيح فإن G10.3 تصبح I 10 ولذا تطبع 2345 مضيطة من اليمين في الأعمدة من 1 إلى 10 . وحيث أن K متغير منطقي ، فإن G10.3 تصبح L10 ، ولذا تطبع F في العمود 20 .

١٠ - ٨ افرض أن A مجموعة متراصة خطية مكونة من سبعة عناصر تحتوي على مايل على الترتيب :

111.18, -22227000.0, -6.15, 0.0000033333, 44.4, -0.023, 888.88

أوجد الخرج التالي :

```
DO 100 K = 1, 7
    WRITE(6, 40) A(K)
40    FORMAT(1X, G12.4)
100 CONTINUE
```

تقرب G12.4 كل رقم إلى أربعة أماكن معنوية كما يلي :

111.2, -22230000.0, -6.150, 0.000003333, 44.40, -0.02300, 888.9

سوف تظيح في الشكل الأسى الأرقام التي لا يمكن طباعتها في حقول ذات عرض 8 = 4 - 12 = 4 - w (أى التي تكون قيمتها المطلقة كبيرة أو صغيرة نسبياً) . وسوف تظيح الأرقام الأخرى في شكل F- مضبطة من اليمين ومنتهية بأربع مسافات كالتالي :

```
. 111.2
-0.2223E 08
-6.150
0.3333E-05
44.40
-0.02300
888.9
```

معامل المقياس التدريجي

١٠ - ٩ أوجد القيمة الداخلية إذا أعطيت أكواد الصيغ الآتية وكذا الرقم الخارجى في المدخلات .

الرقم الخارجى كمدخل	أكواد الصيغ	
123.45	3PF6.1	(أ)
123.45 (٢)	-4PF6.1	(ب)
123.45	4PE6.1	(ج)
123.45E1	-3PE8.1	(د)
12345.	1PF6.2	(هـ)
12345E - 2	-1PE9.2	(و)

تأثير معامل التدرج SP على الأرقام الخارجية كمدخل مكتوب من غير اس وفيما يلي :

$$\text{الرقم الداخلى} = \text{الرقم الخارجى} \times 10^{-s}$$

لا يؤثر معامل التدرج على الأرقام المكتوبة بأس كما يلي (d) و (f) من ثم ،

(أ) 0.12345 (ب) 1234500 (ج) 0.012345 (د) 1234.5 (هـ) 1234.5E1 أو 1234.5E1 (و) 12345

(و) تضاف، علامة عشرية بين 3 ، 4 لتعطى 2 - 123.45E أو 1.2345 . ليس لمعامل التدرج أثر على الرقم .

١٠ - ١٠ : افرض أن 12345.6 هو الرقم الداخلى . أوجد الخرج إذا كانت مواصفات الحقل هي :

(أ) 2PF12.1 (ب) -3PF12.2
(ج) -1PF12.2 (د) 2PE12.4
(هـ) -1PE15.6 (و) 1PE15.3

تذكر أولاً أن $Fw.d$ تقرب الرقم إلى عدد d من الأماكن العشرية بينما تقرب $Ew.d$ الرقم إلى عدد d من الأرقام المنوية .

بالنسبة لحقل F فيضرب معامل التدرج sP القيمة الداخلية في 10^s من ثم ، تصبح لدينا ما يأتي :

(أ) 1234560.0 (ب) 12.35 (ج) 1234.56

بالنسبة لحقل E فلا يغير معامل التدرج قيمة الرقم ، كما يحدث مع حقل F ولكن تغير في الشكل الأسى فقط . وبالتحديد ، تحرك العلامة العشرية بضرب الجزء الأساسى للصورة الأسية في 10^s ثم تضاف s — إلى الأس ويصبح لدينا مايل :

() 12.35E + 03 (هـ) 0.0123456E + 06 (و) 1.23E + 04

١٠ - ١١ : افرض أن A تحتوى على 111.888 أوجد الخرج إذا كان الأمر هو :

WRITE(6, 10) A, A, A, A, A
10 FORMAT(1X, F10.2, E12.4, 1PF10.2, E12.4, F10.2)

يطبق معامل التدرج sP على مواصفات كل حقل تال حتى يتم مقابلة معامل تدرج آخر . من ثم ، يكون الخرج كما يلي :

111.89 0.1119E + 03 1118.88 1.119E + 02 1118.88

حيث تطبع الأرقام مضبوطة من اليمين في حقول ذات عرض 10 ، 12 ، 10 ، 12 ، 10 على الترتيب .

١٠ - ١٢ : افرض في المسألة السابقة ١٠ - ١١ ، أن المطلوب هو تطبيق معامل التدرج على A الثالثة وليس على الرابعة والخامسة . كيف

تكتب جملة ' T FORM ؟

تكتب 0P عندما لانرغب في تطبيق معامل التدرج على أى حقل تال بعد ذلك . من ثم ستكتب جملة FORMAT كالتالى :

10 FORMAT(1X, F10.2, E12.4, 1PF10.2, 0PE12.4, F10.2)

صيغة (FORMAT) وقت التنفيذ :

١٠ - ١٣ : باعتبار أن سعة الحرف هي $M = 4$ صف المدخل التالى :

INTEGER FORM(6)
DATA FORM/'(1X,', 'I2.', '5X.', 'F6.', '2)', '')

تخزن المجموعة المتراسة FORM كما يلي :

FORM(1)	(1	X	,
FORM(2)	I	2	,	
FORM(3)	5	X	,	
FORM(4)	F	6	.	
FORM(5)	2)		
FORM(6)				

لاحظ أن FORM (6) تحتوي على مسافات (فراغات) فقط .

١٠ - ١٤ افترض أن كلا من J و R مجموعة متراسة خطية بها أربعة عناصر وتخصص لها القسم التالية :

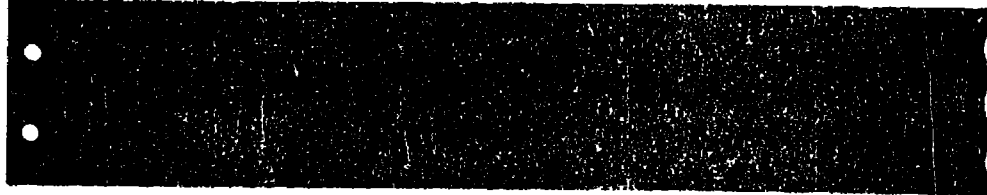
11, 2222, 77, 666 and 3333.33, 44.44, 5.555, 88888.888

على الترتيب . أوجد الخرج إذا نفذنا الآتي باستخدام المجموعة المتراسة FORM في المسألة السابقة ١٠ - ١٣ .

```
DATA K1/'I4./, K2/'F9./
DO 100 I = 1, 4
  IF(J(I).GE.100) FORM(2) = K1
  IF(R(I).GE.1.0E3) FORM(4) = K2
  WRITE(6, FORM) J(I), R(I)
100 CONTINUE
```

سوف يتم عرض الحقل من 2 إلى 4 إذا كانت $J(I) \geq 100$ ويمتد عرض الحقل من 6 إلى 9 إذا كانت $R(I) \geq 1000$ من ثم سيظهر الخرج كما في الشكل ١٠ - ٤ . وفي كل الحالات ، تقرب قيمة R إلى مكانين عشريين .

1234567890123456789012345



شكل ١٠ - ٤

الرسم البياني

١٠ - ١٥ افترض أننا نريد أن نستخدم حلقة DO لنحصل على قيم x من رقم صحيح J إلى رقم صحيح K بمعامل زيادة 1/N حيث N هي رقم صحيح موجب . أوجد (أ) عدد قيم x (ب) قيمة x بدلالة دليل INDEX حلقة DO (ج) قيمة INDEX ، إن

وجدت، التي تغطي محور y . اختبر النتائج في حالة $J = -2$ و $K = 8$ و $N = 4$ أى ، حيث تأخذ x القيم من -2 إلى 8 — بمعامل زيادة قدره $1/4$ أى 0.25 :

(أ) هناك عدد r — K وحدة مدى بين J و K وعدد N من قيم x تناظر كل من هذه الوحدات وهناك أيضاً قيمة ابتدائية L و بذلك يكون هناك مجموع (إجمال) لقيم x قدره :

$$N*(K - J) + 1$$

في المثال ، تأخذ x ! : —

$$-2, -1.75, -1.50, \dots, 0, \dots, 7.75, 8$$

هناك قيم إيجابية ل x قدرها $4(8 + 2) + 1 = 41$

(ب) وقبيل العلاقة بين INDEX و x :

x :	J	$J + 1/N$	$J + 2/N$	$J + 3/N$...	K
INDEX:	1	2	3	4	...	$N*(K - J) + 1$

لاحظ أن :

$$x = J + (INDEX - 1)/N$$

في المثال .

$$x = -2 + (INDEX - 1)/4$$

(ج) يظهر محور y عند $x = 0$ ولكن x يمكن أن تساوى صفرًا فقط إذا كانت $J \leq 0 \leq K$ ، في هذه الحالة ، هناك عدد $N*(-J) + 1$ من قيم x تسبق $x = 0$ وبذلك ، يظهر محور y عندما يكون $INDEX = N*(-J) + 1$ يظهر محور y في هذا المثال عندما يكون $INDEX = 4(2) + 1 = 9$.

١٠ - ١٦ افترض أن كل قيم y تقع بين الرقين الصحيحين L و M (حيث $L < M$) أوجد دالة مقياس الرسم التي تدرج كل قيمة ل y إلى عدد صحيح J بين 1 و 101 . (نعتبر هنا أن عدد أعمدة الرسم البياني 101) اختبر دالة مقياس الرسم عند $L = -10$ و $M = 30$ أى ، حيث قيم y تقع بين -10 ، 30 .

بإضافة L — إلى لا نحصل على قيم بين 0 ، $M - L$. و بالضرب في $100/(M - L)$ نحصل على قيم بين 0 و 100 . وجمع 1 عن النتيجة نحصل على قيم بين 1 و 101 . وحيث أننا نريد أن نقرب إلى عدد صحيح J يقع ما بين 1 و 101 فضلاً عن البتر ، يجب أن نجمع 0.5 أيضاً . وبذلك تكون دالة مقياس الرسم كما يلي :

$$JSCALE(Y) = INT((Y + FLOAT(-L))*(100.0/FLOAT(M - L)) + 1.5)$$

في المثال :

$$JSCALE(Y) = INT((Y + 10.0)*(100.0/40.0) + 1.5)$$

١٠ - ١٧ اكتب البرنامج الذي يرسم القطع المكافئ بيانياً :

$$y = 3x^2 - x - 8$$

من $x = -3$ إلى $x = 3$ افترض أن y تقع ما بين -10 ، 30 واحسب y لقيم x التي تتغير بمقدار 0.1 .

تستخدم حلقة DO (بالدليل I) لقيم x المختلفة . وفيما يلي العلاقة بين I و x :

x: -3 -2.9 -2.8 ... 0 ... 2.9 3
I: 1 2 3 ... 31 ... 60 61

لاحظ أن هناك 61 قيمة لـ x ، يظهر محور y عند I = 31 ويمكن أن نحصل على x من I بواسطة :

$$x = -3.0 + \text{FLOAT}(I - 1) * 0.1$$

(أنظر مسألة ١٠ - ١٥)

بفرض أن رسمنا البياني سيحتوي على 101 عمود ، يمكن أن تدرج كل قيمة لـ y إلى عدد صحيح ل يقع ما بين 1 و 101 باستخدام دالة مقياس الرسم التالية :

$$\text{JSCALE}(Y) = \text{INT}((Y + 10.0) * (100.0 / 40.0) + 1.5)$$

وكما سبق أن ناقشنا ذلك في المسألة ١٠ - ١٦ . لاحظ أن $\text{JSCALE}(0.0) = 26$ لذا يكون العمود 26 من الرسم البياني هو محور x .

وفيما يلي برنامج الفورتران الذي يرسم القطع المكافئ . الإجراء هو نفس الإجراء الذي تمت مناقشته في قسم ١٠ - ٨ . وبالكتابة : سيكون هناك سطر من النقاط (.....) لكل محور ، وحرف x لكل نقطة من الرسم .

```
C      GRAPH OF A PARABOLA
      INTEGER DOT, CROSS, BLANK, LINE(101), LINEY(101)
      DATA DOT, CROSS, BLANK, LINE, LINEY/'.', 'X', '.', '101*', '101*.'/
C      BEGIN NEW PAGE
      WRITE(6, 10)
10     FORMAT(1H1, 10X, 'GRAPH OF A PARABOLA'///3X, 'X', 6X, 'Y'//)
      DO 100 I = 1, 61
          X = -3.0 + FLOAT(I - 1) * 0.1
          Y = 3.0 * X * X - X - 8.0
          J = INT((Y + 10.0) * (100.0 / 40.0) + 1.5)
          IF (I.EQ.31) GO TO 50
          LINE(J) = DOT
          LINE(J) = CROSS
          WRITE(6, 20) X, Y, LINE
20     FORMAT(1X, 2(F4.2, 2X), 101A1)
          LINE(J) = BLANK
          GO TO 100
50     LINEY(J) = CROSS
          WRITE(6, 20) X, Y, LINEY
100    CONTINUE
      STOP
      END
```

١٠ - ١٨ اكتب البرنامج الذي يقرأ رقماً صحيحاً N ويطبع عدداً قدره N^2 من النجوم بحيث يكون هناك عدد N من الأسطر لكل منها عدد N نجمة في الأعمدة 1 و 3 و 5 و ... و $2N - 1$ يجب أن تبدأ الصورة في السطر الخامس من صفحة جديدة .

نحجز أولاً مجموعة متراصة خطية LINE بها 132 عنصر (يناظر كل عنصر منها عمود على صفحة الطباعة) ، وبدء ذلك تخزين مساحة في كل عنصر من LINE . بعد قراءة N ، نخزن حيثنذ نجمة في العناصر LINE(1) و LINE(3) و ... و LINE (2N - 1) وفيما يلي البرنامج :

```

C      PROGRAM PRINTING A SQUARE OF ASTERISKS
      DIMENSION LINE(132)
      INTEGER AST
      DATA LINE/132*' ', AST/'*'/
      READ(5, 10) N
10     FORMAT(I5) .
      NN = 2*N - 1
      DO 100 K = 1, NN, 2
          LINE(K) = AST
100    CONTINUE
C      SKIP 5 LINES ON NEW PAGE
      WRITE(6, 20)
20     FORMAT('1'////)
C      PRINT THE ARRAY N TIMES USING A-FIELD
      DO 200 K = '1, N
          WRITE(6, 30) LINE
30     FORMAT(1X, 132A1)
200    CONTINUE
      STOP
      END

```

مسائل تكميلية

جمل DATA

١٠-١٩ اكتب جملة DATA لإعطاء قسم ابتدائية كما يلي :

(أ) $J = 5$ و $K = 5$ و $L = 5$ و $M = .TRUE.$ (حيث M متغير منطقي)(ب) $A(I, J) = 0$ لكل عنصر من المجموعة المتراسة $A(4 \times 5)$.

١٠-٢٠ أوجد المدخلات فيما يلي :

DATA L, M, N/2*2, '2*2', X, Y, Z/2.5, 2*3.5/

DIMENSION X(6)

DATA X, Y, Z/5*2.22, 2*3.3, 4.4/

LOGICAL K(4)

DATA J, K, L/ 555, 3*.FALSE., .TRUE., 'TRUE'/

(أ)
(ب)

(-)

حقل T-

١٠-٢١ افرض أنه تم تثقيب بطاقة بيانات كالآتي :

1										2										3									
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5					
NOW AND THEN										4 4 4 4 5 5 5 5 5																			

أوجد المدخلات لو نفذنا الآتي :

LOGICAL J, K	(-)	(a)	READ(5, 10), A, B	(أ)
READ(5, 30) I, J, K, C, D			10 FORMAT(T16, 2F5.2)	
30 FORMAT(T16, I3, T4, 2L4, T16, 2F3.1)		(b)	READ(5, 20) X, J, K	(ب)
			20 FORMAT(T14, F4.2, 2I3)	

١٠ - ٢٢ أوجد المخرج لكل جزء برنامج ما يلي :

(ب) $A = 22.22$ $J = 1111$ (١)
 $B = 77.77$ $K = 5555$
 $C = 11.11$ $L = 8888$
 $WRITE(5, 20) A, B, C$ $WRITE(5, 10) J, K, L$
 $20 \text{ FORMAT}(T16, F5.1, 5X, F5.1, T1, F5.1)$ $10 \text{ FORMAT}(T16, I5, T6, I5, T26, I5)$

حقل - G :

١٠ - ٢٢ إفرض أنه تم تنقيب بطاقة بيانات كالآتي :

```

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 .
1 1 1 1 4 4 4 4 6 6 6 E 2 THAT
    
```

أوجد المدخل لو نفذنا الآتي :

(ب) LOGICAL J, K (أ) LOGICAL J, K
 $READ(5, 20) X, N, Y, J, K$ $READ(5, 10) N, X, Y, J, K$
 $20 \text{ FORMAT}(G7.1, 4G4.2)$ $10 \text{ FORMAT}(5G5.2)$

١٠ - ٢٤ أوجد المخرج إذا طبعت قيم J و N و K و X و Y المسألة السابقة ١٠ - ٢٣ باستخدام مايلي :

$WRITE(6, 30) N, J, K, X, Y$
 $30 \text{ FORMAT}(1X, 5G10.3)$

١٠ - ٢٥ إفرض مجموعة متراصة خطية A بها خمسة عناصر تحتوي بالترتيب على مايلي :

$44.444, 0.66666 \times 10^{-6}, 33.3 \times 10^2, 22.2 \times 10^8, 333.3 \times 10^{-2}$

أوجد المخرج إذا كان الأمر هو مايلي :

$DO 100 K = 1, 5$
 $WRITE(6, 50) A(K)$
 $50 \text{ FORMAT}(11X, G12.4)$
 100 CONTINUE

معامل التدرج :

١٠ - ٢٦ أوجد القيمة الداخلية إذا أعطيت كود الشكل (مواصفات الحقل) والرقم الخارجي كدخلات :

(أ)	(د)	(ج)	(ب)	(أ)	
$-1PE10.1$	$-4PE10.1$	$3PE5.1$	$-3PF5.1$	$2PF5.1$	كود الشكل : المدخل :
$7777E + 2$	$77.77E + 2$	77.77	77.77	77.77	

١٠ - ٢٧ إفرض أن الرقم الداخل هو 444.888 أوجد المخرج إذا كانت مواصفات الحقل كل يل :

$-1PE15.5$ (أ)	$-PF10.3$ (ج)	$3PF10.2$ (أ)
$2PE15.4$ (د)	$4PE15.3$ (د)	$-2PF10.1$ (ب)

١٠ - ٢٨ افترض أن X تحتوي 444.777 . أوجد الخرج إذا كان الأمر هو مايل :

```
WRITE(6, 10) X, X, X, X, X, X
10 FORMAT(1X, F12.2, 3X, E12.4, 3X, 2PF12.2, 3X, E12.4, 3X, OPF12.2, 3X, E12.4)
```

قاعدة الأقواس الهرى

١٠ - ٢٩ إدريس جزء البرنامج التالى :

```
DIMENSION K(6)
DATA K/4*33, 2*66/
WRITE(6, 10) K
```

أوجد الخرج إذا كانت جملة FORMAT المصاحبة هى :

```
10 FORMAT(1X, I2, 2(3X, I2), 8X, I2) (أ) 10 FORMAT(1X, I2, (I4, I6)) (ب)
```

١٠ - ٣٠ إدريس جملة READ التالية :

```
READ(5, 20) A, B, C, X, Y, Z
```

واكتب جملة FORMAT بحيث :

تقرأ A من أول بطاقة باستخدام F10.2

وتقرأ B و C و X من ثانى بطاقة باستخدام F10.3 و F10.2 و F10.4

وتقرأ Y و Z من ثالث بطاقة باستخدام F10.3 ، F10.2

صمغ (FORMAT) و لت التنفيذ :

١٠ - ٣١ بفرض أن سمة الحرف $M = 4$ صف المدخل :

```
INTEGER FORM(5)
DATA FORM/'(1X,', 'I3,', '3X,', 'F6', '.3)'/
```

١٠ - ٣٢ افترض أن J و X مجموعات متراسة خطية كل منها بها خمسة عناصر. ، وافترض أن J و X قد أعطيت القيم التالية :

333, 33333, 33, 333333, 3 and 4.4, 66666.66, 1.1, 88.88888, 2222.2

عل الترتيب . أوجد الخرج إذا نفذنا الآق باستخدام المجموعة المتراسة FORM فى المسألة السابقة ١٠ - ٣١ :

```
DATA IA, IB/'I7,', 'F10'/
DO 100 K = 1, 5
  IF(J(K).GE.1000) FORM(2) = IA
  IF(X(K).GE.100.0) FORM(4) = IB
  WRITE(6, FORM) J(K), X(K)
100 CONTINUE
```

١٠ - ٣٣ افترض أن مجموعة متراسة خطية J تحتوي 75 رقاً صحيحاً موجباً أقل من 10.000 . أكتب جزء البرنامج الذى يطبع الأرقام

فى عمود بحيث يكون مضطماً من اليسار ، كما يلى مثلا :

28
111
5
3344
62
⋮

الرسم البياني

١٠ - ٢٤ اكتب البرنامج الذي يرسم الدالة $y = \sin x$ بالنسبة إلى $10 \leq x \leq 10$ (تذكر أن $1 \leq y \leq -1$ لأي قيمة لـ x)

١٠ - ٢٥ افرض أن القيم y_1 و y_2 و ... و y_N خزنت مجموعة متراصة Y بها N عنصر . اكتب البرنامج الفرعي SUBROUTINE

ENDS(Y, N, YMIN, YMAX)

الذي يجد القيمة الصغرى YMIN والقيمة العظمى YMAX للعناصر في المجموعة المتراسة Y .

١٠ - ٢٦ اكتب البرنامج الذي يرسم الدالة :

$$y = x^3 + 2x^2 - 15x - 8$$

من $x = -5$ إلى $x = 4$ حيث تتغير قيم x بمقدار 0.2 . بالتحديد خزن كل قيم y أولاً في مجموعة متراصة خطية و بعد ذلك استخدم البرنامج الفرعي ENDS في المسألة ١٠ - ٢٥ ودالة مقياس الرسم المشابهة إلى الدالة المشروحة في المسألة ١٠ - ١٦ لتدريج y إلى رقم صحيح J يتراوح بين 1 و 101 .

١٠ - ٢٧ اكتب البرنامج الذي يقبل رقمين صحيحين موجبين J و K ويطبع مستطيل $J * K$ من النجوم بحيث أن تكون هناك عدد L من الأسطر بعدد K من النجوم تظهر في الأعمدة 1 و 3 و 5 و ... و $2K - 1$ (يجب أن تبدأ الصورة في السطر الثالث من صفحة جديدة) .

اجابات للمسائل التكميلية المختارة

١٠ - ١٩

(أ) DATA J, K, L, M/3*5, .TRUE./ (ب) DATA A/20*0.0/

١٠ - ٢٥ (أ) تحتوي L و M على 2 وتحتوي N على الحروف الأربعة $2 * 2$ باعتبار أن السمة $M = 4$ (، وتحتوي X على 2.5 وتحتوي Y و Z على 3.5) .

(ب) تحتوي $X(1)$ و ... $X(5)$ على 2.22 وتحتوي $X(6)$ و Y على 3.3 وتحتوي Z على 4.4 .

(ج) تحتوي J على 555 وتحتوي $K(1)$ و $K(2)$ و $K(3)$ على FALSE. وتحتوي $K(4)$ على TRUE. وتحتوي L على سلسلة الحروف TRUE « باعتبار أن السمة $M = 4$ » .

١٠-٣١ يخزن المجموعة المتراسة FORM كما يلي :

FORM(1)	(1	X	,	
FORM(2)	I	3	,		
FORM(3)	3	X	,		
FORM(4)	F	6			
FORM(5)	.	3)		

١٠-٣٢ أنظر شكل ١٠-٦



شكل ١٠-٦

١٠-٣٣ تلميح : استخدم صيغ (FORMAT) وقت التنفيذ مع كود الشكل I_1 و I_2 و I_3 و I_4 تبعاً لقيمة J التي تحتوي واحد أو اثنين أو ثلاثة أو أربعة أرقام على الترتيب .

الفصل الحادي عشر

ملاح متنوعه للفورتران

١ - ١١ مقدمة

يتناول هذا الفصل خصائص إضافية متعددة للغة البرمجة فورتران . وبالتحديد ، سوف ندرس بصورة عامة جمل النوع وجملة IMPLICIT ومتغيرات متضاعفة الدقة DOUBLE PRECISION ومتغيرات مركبة COMPLEX وجمل المتغيرات المشتركة COMMON والمتكافئة EQUIVALENCE .

٢ - ١١ جمل النوع (TYPE)

هناك أنواع متعددة من المتغيرات حقيقية REAL وصحيحة INTEGER ومنطقية LOGICAL ومتضاعفة الدقة DOUBLE PRECISION ومركبة COMPLEX . سبق وناقشنا المتغيرات الحقيقية REAL والصحيحة INTEGER في الفصل الثاني ، وتمت مناقشة المتغيرات المنطقية LOGICAL في الفصل التاسع ، وسوف نناقش في هذا الفصل المتغيرات المركبة ومتضاعفة الدقة COMPLEX و DOUBLE PRECISION في الأقسام ١١ - ٤ و ١١ - ٥ .

يسمح لنا الفورتران باستخدام « الحرف الأول » كتقليد يدل على نوع المتغير وبالتحديد ، نأى متغير يبدأ اسمه بأى حرف من الحروف I أو J أو K أو L أو M أو N فهو متغير صحيح أما إذا ابتداء بأى حرف آخر فهو متغير حقيقي . ومع ذلك ، فيمكن التغلب على هذا التقليد باستخدام جملة النوع (Type) .

ويكون لجملة النوع (Type) الشكل التالي :

قائمة متغيرات TYPE

حيث تفصل أسماء المتغيرات بفواصلات . وتعلمن جملة النوع عن نوعية المتغيرات في القائمة التالي :

REAL MONEY, RATE, NEW
INTEGER X, Y, NEXT
LOGICAL A, B
DOUBLE PRECISION S, T
COMPLEX ROOT, COEF

نحن أن MONEY و RATE و NEW متغيرات حقيقية ، وأن X و Y و NEXT متغيرات صحيحة ، وهكذا . لاحظ أنه ليس من الضروري تصنيف RATE في الجملة الأولى أو NEXT في الجملة الثانية ، يشمر بعض المبرمجين في الحقيقة أنها ممارسة جيدة أن نذكر كل المتغيرات في جملة نوع بغض النظر عن حرفه الأول .

يمكن أن تظهر أيضاً أسماء المجموعات المتراسة في جمل النوع (type) . في الحقيقة ، يمكن أن نتجنب استخدام جملة الأبعاد DIMENSION منفصلة تضمين حجم المجموعة المتراسة في جملة النوع فثلا :

INTEGER COUNT(30)
REAL NUMBER(4, 5), LAST

تلمن أن COUNT مجموعة متراسة خطية بها 30 عنصراً وأن NUMBER مجموعة متراسة حقيقية (4 × 5) .
جمل النوع (type) جمل غير منفذة ولذا يجب أن تظهر في بداية البرنامج قبل جملة DATA وقبل أى استخدام للمتغيرات .

١١ - ٣ جملة IMPLICIT

يمكن أن نتوسع في استخدام التقليد للنوع أى « الحرف الأول » باستخدام جملة IMPLICIT . وسوف نوضح ذلك بمثال . ادرس جملة IMPLICIT التالية :

IMPLICIT INTEGER(B, D, W - Z), LOGICAL(A, N - P), COMPLEX(F - H)

هذه الجملة تختبر المترجم أن كل أسماء المتغيرات التي تبدأ بالحروف B و D و W و X و Y و Z متغيرات صحيحة ، وأسماء المتغيرات التي تبدأ بالحروف A و N و O و P هي متغيرات منطقية ، وأسماء المتغيرات التي تبدأ بالحروف F و G و H هي متغيرات مركبة إلا إذا نص على غير ذلك صراحة في جمل نوع . لاحظ أن مدى الحروف يحدد بكتابة أول وآخر حرف منفصلين بإشارة ناقص . لاحظ أيضاً أن الحروف ومداهما يتبع النوع TYPE ويفصولة بفواصل ومحاطة بين أقواس .

ويجب أن نذكر الآق عند استخدام جملة IMPLICIT :

١ - عادة يمكن أن تظهر جملة IMPLICIT واحدة في أى برنامج أو في أى برنامج فرعى . ويجب أن تكون أول جملة ممكنة ، بمعنى أنها يجب أن تكون أول جملة في البرنامج الرئيسى . ويجب أن تأتى في البرنامج الفرعى بعد جملة تعريف البرنامج الفرعى مباشرة .

٢ - يمكن أن نتغلب على جملة IMPLICIT بجملة نوع . علاوة على ذلك ، يبق تقليد النوع العادى أى « الحرف الأول » لأى متغيرات لم تظهر في جملة IMPLICIT أو في جمل نوع . كثال . افرض برنامجاً به الجمل التالية :

IMPLICIT DOUBLE PRECISION (A - D, S), INTEGER(F - G)
COMPLEX ALPHA, BETA
INTEGER SET, CLASS

إذن ALPHA و BETA متغيرات مركبة و SET و CLASS متغيرات صحيحة في ضوء جمل النوع . ANSWER و SUM متغيرات متضاعفة اللقة DOUBLE PRECISION في ضوء جملة IMPLICIT . من ناحية أخرى RATE و KOUNT لم يتضمنا في أى من الجمل ، ولذا فهما متغيرات حقيقية REAL وصحيحة INTEGER على الترتيب .

٣ - معاملات البرنامج الفرعى وأيضاً اسم البرنامج الفرعى FUNCTION يتم تنطيتها في جملة IMPLICIT في البرنامج الفرعى .

١١ - { الدقة المتضاعفة DOUBLE PRECISION

يمكن أن يخزن الحاسب عادة سبعة أو ثمانية أرقام معنوية في اى خلية ذاكرة وسوف تحتفظ حساباته (وتسمى أيضاً حسابات الدقة المفردة) برقم مماثل من الأرقام المعنوية . ومع ذلك ، يسمح لنا الفورتران بالحصول على نتائج أكثر دقة باستخدام ما يسمى قيم وحسابات الدقة المتضاعفة . في هذه الحالة تخزن كل قيمة في مكانين (2) من أماكن الذاكرة ، وعلى ذلك يضاف تقريباً عدد الأرقام المعنوية . (فثلا ، تحتفظ سلسلة IBM360/370 بسبعة أرقام معنوية تحت الحساب المادى ولكن تحتفظ بعدد 16 رقماً معنوياً تحت حساب الدقة المتضاعفة) . سنناقش في هذا القسم ، القوانين الخاصة بحساب الدقة المتضاعفة .

ملاحظة : حيث أن قيم الدقة المتضاعفة تحت عدداً مضاعفاً من أماكن الذاكرة وحيث أن الحساب يأخذ ، وقتاً أطول في التنفيذ ، يتراوح هذا الوقت ما بين مرتين إلى عشر مرات ، لذلك يلزمنا أن نستخدم الدقة المتضاعفة في الحالة التي تتطلب هذه الدقة فقط .

متغيرات

نعلن عن المتغيرات والمجموعات المترابطة الخاصة بالدقة المتضاعفة بواسطة جملة DOUBLE PRECISION كما ناقشنا في قسم ١١-٢ .

ثوابت

تكتب ثوابت الدقة المتضاعفة بطرق مشابهة للثوابت الحقيقية مفردة الدقة ولكن يجب أن توجد علامة عشرية في أى ثابت دقة متضاعفة .

١ - بدون أس . يفسر الثابت الحقيقى المكتوب بدون أس كثابت بدقة مفردة ، إلا إذا تجاوز عدد أرقامه المعنوية العدد الأقصى المسموح به لثابت حقيقى بدقة مفردة . فثلا بالنسبة لسلسلة IBM360/370 التي تسمح بسبعة أرقام معنوية ، سيعتبر الرقم 123.465 ثابت بدقة مفردة بينما سيعتبر الرقم 123.456000000 ثابت بدقة متضاعفة .

٢ - ذات أس . تمثل ثوابت الدقة المفردة في الشكل الأسى باستخدام الحرف E فثلا 46.8086E7 ثابت بدقة مفردة . وتمثل ثوابت الدقة المتضاعفة في الشكل الأسى باستخدام الحرف D بدلا من الحرف E وبذلك ، تمثل 46.8086D7 نفس الرقم بالدقة المتضاعفة ، ومع ذلك ، فإنه يخزن داخلياً بعدد 16 رقماً معنوياً بالكامل . كما يلي :

$$0.4680860000000000 \times 10^9$$

(أى تضاعف أصفار للحصول على 16 رقماً صحيحاً تماماً كما هو المطلوب) .

يجب أن نتذكر أنه لا توجد أرقام صحيحة بدقة متضاعفة .

إدخال / إخراج

يستخدم حقل D لإدخال / وإخراج قيم الدقة المتضاعفة بعيداً عن حقيقة أن حقول D تسمح بتناول أماكن عشرية . أطول ، إلا أن استخدام حقل D مشابه تماماً استخدام حقل E فثلا ، مواصفات الحقل لها الشكل .

Dw.d

حيث w هي عرض الحقل و d هي عدد الأرقام العشرية ولذلك ننصح القراء بمراجعة الفصل الثالث الخاص بتصنيف حقول—E لناقشة هادئة .

عمليات حسابية

تنفذ حسابات الدقة المتضاعفة عندما يكون طرفاً المعادلة من نوع الدقة المتضاعفة ، وتسمح معظم المبرجات بمزيج من حدود الدقة المفردة والدقة المتضاعفة . في مثل هذه الحالة ، يحول حد الدقة المفردة أولاً إلى دقة متضاعفة . (بإضافة أصفار خلف الرقم) وبعد ذلك تجرى حسابات الدقة المتضاعفة . من ناحية أخرى ، لا تسمح معظم المبرجات بمزيج من الأنواع الصحيحة والدقة المتضاعفة ، فيما عدا أنه يمكن أن نرفع قيمة بدقة متضاعفة إلى أس صحيح . نذكر هنا أيضاً أن أى قيمة سالبة بدقة متضاعفة لا يمكن أن ترفع إلى قوة حقيقية أو قوة بدقة متضاعفة كما هو الحال مع الأرقام السالبة الحقيقية (انظر صفحة ٢٨) .

جمل تفصيل

وكما في حالة الدقة المفردة ، فيمكن للأنواع على جانبي إشارة = أن تكون مختلفة . وتحول القيمة التي سوف تخزن دائماً إلى نوع مكان التخزين المناظر . بالإضافة إلى ذلك فإن قيم الدقة المتضاعفة تبتز ولا تقرب ، عند تحويلها إلى دقة مفردة . ادرس سلا الجزء التالى :

DOUBLE PRECISION A

$$A = 1.0$$

$$B = 123.456789098$$

$$J = 123.456789098$$

$$K = 3.4444D3$$

وعلى ذلك

- ١ - يحول الثابت (1.0) إلى دقة متضاعفة ويخزن في A .
- ٢ - يخزن الرقم 123.4567 فقط في B أى ، يبتز الثابت بعد سبعة أرقام مئوية .
- ٣ - يخزن الرقم 123 فقط في J أى ، يبتز الثابت عند العلامة العشرية .
- ٤ - رغم أن 3.4444D3 تساوى 3444.4 ولكن تخصص 3 فقط إلى K حيث يبتز الثابت عند العلامة العشرية .

الدوال المكتبية

تحول الدوال المكتبية DFLOAT و DBLE القيم الصحيحة وقيم الدقة المفردة إلى قيم دقة متضاعفة على الترتيب وتحول الدالة SNGL قيمة بدقة متضاعفة إلى دقة مفردة . هناك أيضاً دوال مكتبية تسمح دائماً بإجراء حسابات الدقة المتضاعفة لمعظم الدوال المستخدمة . فمثلاً ، إذا كانت DX خلاصة دالة بدقة متضاعفة فسوف تعطى :

DSQRT(DX)

الجذر التربيعى لـ DX باستخدام حساب الدقة المتضاعفة . بالمثل تعطى DABS(DX) و DLOG(DX) و DSIN(DX) القيمة المطلقة واللوغاريتم ، وجيب الزاوية لمتغير DX بدقة متضاعفة . وبصورة عامة فالحرف D إذا وضع أمام أى دالة فهو يدل على حسابات الدقة المتضاعفة . وسوف تعطى قائمة بدوال أخرى كثيرة فى الملحق أ وسوف نضطر إلى استخدام :

. DLOG(DBLE(X))

لحصول على اللوغاريتم الطبيعي بالدقة المتضاعفة لقيمة صحيحة بدقة مفردة X حيث أن خلاصة الدالة DLOG يجب أن تكون بدقة متضاعفة .

برامج فرعية

يمكن أن تنقل قيم الدقة المتضاعفة من وإلى برنامج فرعى عن طريق الخلاصات (تماماً مثل أى أنواع أخرى من القيم) ، ومع ذلك ، فيجب أن نعلن عن أسماء هذه الخلاصات وأسماء المعاملات المناظرة لها كتغيرات بدقة متضاعفة في كل من البرنامج الرئيسى والبرنامج الفرعى. يمكن أن نعلن أيضاً أن البرنامج الفرعى FUNCTION نفسه ذو دقة متضاعفة بكتابة DOUBLE PRECISION في بداية جملة تعريف FUNCTION أو بإعلان أن اسم البرنامج NAME ذو دقة متضاعفة بداخل البرنامج الفرعى .
مثلاً فيما يلى ، برنامجان فرعيان متكافئان .

```
DOUBLE PRECISION FUNCTION SUM(A, B)
DOUBLE PRECISION A, B
SUM = A + B
RETURN
END
```

```
FUNCTION SUM(A, B)
DOUBLE PRECISION SUM, A, B
SUM = A + B
RETURN
END
```

وعلاوة على ذلك ، تتطلب معظم المترجمات من البرنامج الداعى أن يعلن عن اسم (NAME) البرنامج الفرعى (الدالة) ذى الدقة المتضاعفة FUNCTION . أى يجب أن تظهر الجملة :

```
DOUBLE PRECISION SUM
```

أوما يكافئها في البرنامج الداعى الذى يستخدم البرنامج الفرعى المذكور أعلاه SUM (وليس هذا المطلب عاماً ، بل يجب على المبرمج أن يبحث عن المتطلبات الدقيقة للمترجم الخاص به) . من ناحية أخرى ، لا يوجد برنامج فرعى بدقة متضاعفة DOUBLE PRECISION SUBROUTINE حيث لا تخصص قيمة لاسم (NAME) البرنامج الفرعى .

مثال ١١ - ١

يفرض أننا استخدمنا سلسلة IBM 260/370 فسوف نحصل من جزء البرنامج التالى :

```
DOUBLE PRECISION DA, DB
DA = 2./3.
DB = 2.0D0/3.0D0
WRITE(6, 10) DA, DA
WRITE(6, 10) DB, DB
10 FORMAT(1X, D23.16, 10X, D12.4)
```

حل :

```
0.6666666000000000D 00      0.6667D 00
0.6666666666666666D 00      0.6667D 00
```

وعذا يحدث لأن 2./3. تحسب بالدقة المفردة وتبتر بعد سبعة أرقام معنوية أى يكون لها القيمة 0.6666666 أما عندما نخزن 2./3. في DA فتحول إلى دقة متضاعفة بإضافة أصفار . من جهة أخرى تحسب 2.0D0/3.0D0 بالدقة المتضاعفة وتبتر بعد 16 رقماً معنوياً ، أى تكون لها القيمة 0.6666666666666666

(ليس هناك تقريب فى هذه الحسابات) ومع ذلك فبندما طبعت هذه القيم تم تقريبها إلى أربعة أرقام معنوية كما وصفت فى الصيغة (FORMAT)

١١ - ٥ الأعداد المركبة

تسمح مترجات الفورتران العمليات الحسابية بتناول أعداد مركبة . دعنا نراجع أولاً حساب الأعداد المركبة وبعد ذلك نناقش طريقة تناول الفورتران لها .

يكتب عادة عدد مركب z فى الشكل :

$$z = a + bi$$

حيث a و b أعداد حقيقية عادية و $i = \sqrt{-1}$. تسمى a الجزء الحقيقى لـ z وتسمى b الجزء التخيل لـ z (يمكن أن نعرف الجزء الحقيقى a فى صورة العدد المركب $(a + 0i)$)

يمكن الحصول على جمع وضرب أعداد مركبة باستخدام القوانين المجمة والموزعة العادية للغير مع اعتبار $i^2 = -1$:

$$(a + bi) + (c + di) = a + c + bi + di = (a + c) + (b + d)i$$

$$(a + bi)(c + di) = ac + bci + adi + bdi^2 = (ac - bd) + (bc + ad)i$$

مرافق العدد المركب $z = a + bi$ يرمز إليه ويعرف بالآق :

$$\bar{z} = a - bi$$

(لاحظ أن $z\bar{z} = a^2 + b^2$. إذا كانت $z \neq 0$ أى ، إذا كانت $a \neq 0$ أو $b \neq 0$ فإنه يمكن إيجاد مقلوب z أى z^{-1})
والقسمة على z مبيته فيما يلى :

$$\frac{w}{z} = wz^{-1} \quad \text{و} \quad z^{-1} = \frac{\bar{z}}{z\bar{z}} = \frac{a}{a^2 + b^2} + \frac{-b}{a^2 + b^2}i$$

حيث w عدد مركب أيضاً . ونعرف أيضاً :

$$w - z = w + (-z) \quad \text{و} \quad -z = -1z$$

قيمة z المطلقة (أو القيمة المطلقة لـ z حيث $z = a + bi$) تكتب بالصورة $|z|$ وتعرف بواسطة :

$$|z| = \sqrt{a^2 + b^2} = \sqrt{z\bar{z}}$$

مثال ١١ - ٢

افترض أن $z = 5 - 3i$ و $w = 5 - 2i$ من ثم يمكن استنتاج التالى :

$$z + w = (2 + 3i) + (5 - 2i) = 2 + 5 + 3i - 2i = 7 + i$$

$$zw = (2 + 3i)(5 - 2i) = 10 + 15i - 4i - 6i^2 = 16 + 11i$$

$$\bar{z} = \overline{2 + 3i} = 2 - 3i \quad \text{and} \quad \bar{w} = \overline{5 - 2i} = 5 + 2i$$

$$\frac{w}{z} = \frac{5 - 2i}{2 + 3i} = \frac{(5 - 2i)(2 - 3i)}{(2 + 3i)(2 - 3i)} = \frac{4 - 19i}{13} = \frac{4}{13} - \frac{19}{13}i$$

$$: |z| = \sqrt{4 + 9} = \sqrt{13} \quad \text{and} \quad |w| = \sqrt{25 + 4} = \sqrt{29}$$

حيث أن العدد المركب يتكون من جزئين ، جزء حقيقي وجزء تخيل ، فإننا نحتاج إلى مكانين في الذاكرة لتخزين العدد المركب . وفيما يلي القواعد المتعلقة بالثوابت والمتغيرات المركبة وطريقة حسابها .

متغيرات

يعلن عن المتغيرات والمجموعات المترابطة المركبة باستخدام جملة **COMPLEX** كما تمت مناقشتها في قسم ١١ - ٢ .

ثوابت

يمثل الفورتران الثابت المركب بزوج مرتب من الثوابت الحقيقية تفصل بينهما فصلة ويحاطا بأقواس . يشير الرقم الحقيقي الأول إلى الجزء الحقيقي من العدد المركب ويشير الرقم حقيقي الثاني إلى الجزء التخيل . يمكن أن يكتب كل من الرقمين بأس أو بدون أس ولكن لا يمكن أن تكون رقماً صحيحاً . وبذلك يمكن كتابته :

$$1 + 2i$$

في الفورتران بأي من الطرق التالية :

(1.0, 2.0)
(0.1E1, 2.0)
(1.0, 2.0E0)
(0.1E1, 0.2E1)

و لكن لا يمكن تمثيلها كما يلي :

$$(1, 2)$$

من ناحية أخرى لا يقبل زوج مرتب من المتغيرات الحقيقية أو التعميرات عند تعريف قيمة مركبة . فمثلاً إذا عرف أن **X** متغير حقيقي فلا يمكن أن نستخدم :

$$(X, 2.0)$$

نتعريف قيمة مركبة . مع ذلك نجد في الفورتران دالة مكتوبة مركبة **CMPLX** تحول زوجاً مرتباً من تعبيرات حقيقية إلى ثابت له قيمة مركبة . بذلك تكون :

$$\text{CMPLX}(X, 2.0) \text{ and } \text{CMPLX}(X + 2.0, X - 2.0)$$

طرق الفورتران لكتابة الأعداد المركبة $x + 2i$ و $(x - 2)i$ و $(x + 2)$

الإدخال / الإخراج

ليس هناك حقل معين لإدخال / إخراج الأعداد المركبة . ومع ذلك حيث أن كل رقمين حقيقيين يناظران عدداً مركباً ، فيجب أن تعطى مواصفات حقلين لكل متغير مركب . فمثلاً :

```
COMPLEX Z
READ(5, 10) Z
10 FORMAT(F10.2, 10X, F10.2)
```

سوف تخصص القيمة فى الأعمدة من 1 إلى 10 كجزء حقيقى لـ Z والقيمة فى الأعمدة من 21 إلى 30 كجزء تخيل لـ Z . وبذلك إذا أعدت بطاقة البيانات كما يلى :

1	2	3
12345678901234567890123456789012345		
12.34	333.33	13.579

سوف تكون قيمة Z هى القيمة $12.34 + 13.579i$. ومن الطبيعى أن تكون مواصفات الحقول إما حقل B أو حقل F أو حقل G أو توافقية منها .

العمليات الحسابية

يجرى الحساب المركب عندما يكون كل من طرفى العملية مركب ، أو عندما يكون أحدهما حقيقياً والآخر مركباً . فى الحالة الأخيرة ، تحول القيمة الحقيقية وتكون a إلى مكافئها المركب $(a, 0.0)$ وبعد ذلك يتم تأدية الحساب المركب . فمثلاً إذا كانت X متخيلية و C مركبة فتعابير الفورتران التالية مقبولة :

$$X + (1.0, 2.0), \quad X + C, \quad C - X, \quad X/C, \quad C*X, \quad C + 1.0$$

إلا كان المترجم يقبل مزيجاً من القيم الصحيحة والحقيقية يمكن أيضاً استخدام القيم الصحيحة مع القيم المركبة ، فمثلاً تكون $C + 1$ منبولة ويتم تأدية الحساب المركب . ومع ذلك هناك قيد واحد خاص يسرى على الأعداد المركبة ولا يسرى على الأرقام الحقيقية وهو : لا يمكن رفع رقم مركب إلى قوة حقيقية . بمعنى آخر فإن العدد المركب يرفع إلى قوة صحيحة فقط .

جمل التخصيص

يمكن أن تخصص قيم حقيقية أو صحيحة أو مركبة إلى متغيرات مركبة . فمثلاً إذا كانت A و B متغيرات مركبة و X و Y متغيرات حقيقية ، فجمل التخصيص الآتية صحيحة :

$$\begin{aligned} A &= (1.0, 0.2E1) \\ A &= A + B**2 \\ A &= 25 \\ A &= 3*X \\ A &= 3*B + Y - 8.0 \\ A &= CMPLX(X, Y + 2.0) \end{aligned}$$

وبالتحديد ، تحول القيم الصحيحة أو الحقيقية إلى قيم مركبة قبل تخزينها فى المتغير المركب A . من ناحية أخرى ، لا يمكن تخصيص قيم مركبة إلى متغيرات حقيقية أو صحيحة وبذلك تكون الجمل التالية :

$$\begin{aligned} X &= A + B \\ K &= B**2 \end{aligned}$$

غير صحيحة .

حيث n و n_1 و n_2 و ... و n_k ثوابت صحيحة بدون إشارة تدل على أرقام جمل (عناوين) و J متغير صحيح . تخصص جملة ASSIGN الثابت n إلى المتغير J . عندما تنفذ جملة GO TO المخصصة ، ينتقل التحكم إلى الجملة المرقمة بالرقم n أى القيمة الحالية لـ J .

تلخص قوانين جملة ASSIGN وجملة GO TO المخصصة كالتالى :

١ - يجب أن يكون المتغير J متغيراً صحيحاً (بدون دليل) .

٢ - لا يمكن تغيير المتغير J إلا بجملة ASSIGN أخرى . فبالجزء التالى :

```
ASSIGN 25 to LAST
LAST = LAST + 5
GO TO LAST, (20, 25, 30, 35)
```

غير مسموح به

٣ - يجب أن تلى قيمة J عند ظهورها فى قائمة أرقام الجمل المتغير J فى جملة GO TO .

١١ - ٧ المداخل والرحوع المتعددة لبرنامج فرعى

(١) مداخل متعددة

عادة ، عندما يظهر اسم برنامج فرعى (SUBROUTINE أو FUNCTION) فى برنامج دافع فالمدخل إلى البرنامج الفرعى هو عند أول جملة منفذة تلى جملة تعريف البرنامج الفرعى . توفر بعض المترجمات الأوامر للدخول لبرنامج فرعى عند نقاط دخول مختلفة . وتستخدم جملة ENTRY فى البرنامج الفرعى للإشارة إلى نقاط الدخول البديلة . ولهذا الجملة الشكل التالى :

ENTRY NAME (قائمة معاملات)

حيث يختلف اسم المدخل (NAME) عن اسم البرنامج الفرعى . جملة ENTRY جملة غير منفذة لذلك توضع فى البرنامج الفرعى عند النقطة التى سوف يتم عندها الدخول . لن يكون للجملة تأثير على منطق البرنامج الفرعى .

يمكن أن يكون هناك عدة جمل تملن عن المداخل ENTRY فى البرنامج الفرعى . كل جملة تعرف نقطة دخول مختلفة . ومن الواضح أن كل نقطة دخول يجب أن يكون لها اسم خاص بها . وعلاوة على ذلك ، لا يستلزم أن تتفق المعاملات اللازمة لنقطة دخول مع معاملات البرنامج الفرعى ، ولا تستلزم أيضاً أن تتفق مع معاملات أى نقطة دخول أخرى . ولكن عندما تستخدم نقطة دخول معينة NAME فى البرنامج الداعى ، يجب أن تتفق قائمة خلاصات جملة النداء مع معاملات جملة ENTRY .

شكل جملة النداء لنقطة دخول فى برنامج فرعى مشابه لجملة النداء الخاصة ببرنامج فرعى SUBROUTINE أو FUNCTION فى كلا الحالتين عندما نستخدم اسم المدخل NAME بواسطة برنامج دافع ، سيتم التنفيذ عند أول جملة منفذة تلى إعلان المدخل ENTRY فى البرنامج الفرعى .

وهيكل البرنامج التالى يوضح المفاهيم السابقة :

برنامج رئيسى	برنامج فرعى
.....	SUBROUTINE ABC(X, Y, Z)
CALL ABC(P, Q, R)
.....	GO TO 10
CALL TWOA(E, F)	ENTRY ONEA(W, K)
.....
CALL ONEA(S, M)	GO TO 10
.....	ENTRY TWOA
STOP
END	10 RETURN
	END

(ب) رجوع متعدد (للبرامج الفرعية SUBROUTINE فقط)

عند مقابلة جملة الرجوع RETURN في أى SUBROUTINE ينتقل التحكم عادة إلى أول جملة منفذة تالية لجملة CALL في البرنامج الداعى . يسمح لنا الفورتران أيضاً بالرجوع إلى نقط مختلفة من البرنامج الداعى . ونوضح هذه الإمكانية باستخدام هيكل البرنامج التالى :

برنامج رئيسى	برنامج فرعى
DIMENSION TEST(100)	SUBROUTINE GRADE(R, *, S, T, *, W)
.....	DIMENSION W(100)
CALL GRADE(A, 30, B, C, 20, TEST)
.....	RETURN 2
END
	RETURN

	RETURN 1
	END

لاحظ أن هناك نجمتين في قائمة المعاملات في جملة تعريف البرنامج الفرعى SUBROUTINE لاحظ أيضاً أرقام الجمل 30 و 20 في قائمة الخلاصات المناظرة لهاتين النجمتين على الترتيب . وتبعاً لذلك فجملة RETURN 1 و RETURN 2 تتناظر أرقام الجمل 30 و 20 على الترتيب . وعلى ذلك إذا قابلت RETURN 1 في البرنامج الفرعى ، فسوف ينتقل التحكم إلى الجملة رقم 30 في البرنامج الداعى ، وإذا قابلنا RETURN 2 فسوف ينتقل التحكم إلى الجملة رقم 20 في البرنامج الداعى . من ناحية أخرى ، عندما يقابل RETURN فسوف يعود التحكم إلى أول جملة منفذة تالية لجملة CALL.

وبصورة عامة فإن جملة RETURN المتعددة لها الشكل :

RETURN n

حيث n ثابت صحيح بدون إشارة . تظهر هذه الجملة فقط في برنامج فرعى به على الأقل عدد n من النجوم في قائمة المعاملات الخاصة به وحيث تقابل كل نجمة رقم جملة في قائمة الخلاصات . بعد ذلك عندما نقابل RETURN n في البرنامج الفرعى فسوف ينتقل التحكم إلى الجملة التى رقمها مساو لرقم n في قائمة الخلاصات في البرنامج الداعى .

١١ - ٨ جمل COMMON غير الميزة

تذكر أن المتغيرات في أى برنامج فرعى خاصة بهذا البرنامج الفرعى ما عدا المعاملات . وبالتالي فإن الوسيلة الوحيدة لتبادل المعلومات بين البرنامج الداعى والبرنامج الفرعى هى تحديدها في قائمة خلاصات . ومع ذلك فيمكن أن يكون هذا التنظيم معقداً إذا كان عدد أماكن الذاكرة التي سوف تنقسمها عدة برامج فرعية كبيرة جداً . يمدنا الفورتران بوسيلة بديلة يمكن البرامج الفرعية من اقتسام مساحات ذاكرة مشتركة . بالتحديد ، يمكن تعريف مساحة تخزين مشتركة باستخدام جملة COMMON سيكون لكل البرامج التي بها تعريف مناسب لهذه الجملة حق التوصل لأي بيانات مخزنة في هذه المساحة .

هناك نوعان من جمل COMMON : COMMON ميرة و COMMON غير مميزة (خالية) . كلا النوعين جمل غير منفذة . ويجب أن توضع في بداية أى برنامج قبل أى جملة منفذة . سنناقش جمل COMMON غير الميزة في هذا القسم . وفي القسم التالي سنناقش جمل COMMON الميزة . وفيما يلي جمل COMMON غير الميزة (أى الخالية) النموذجية :

DIMENSION Y(2, 3), W(2)
COMMON X, Y, Z, W

لاحظ أن جملة COMMON تبدأ بكلمة COMMON وتنتميها قائمة من أسماء المتغيرات ربما يكون فيها أسماء مجموعات مترامه تفصل عن بعضها بواسطة فاصلات . يمكن أن نتجنب استخدام جملة DIMENSION منفصلة للمجموعات المترامه الموجودة في جملة COMMON بتضمين حجم المجموعة المترامه في جملة COMMON نفسها . فثلا كل من التغطيات التالية :

COMMON X, Y(2, 3), Z, W(2) or DIMENSION Y(2, 3)
COMMON X, Y, Z, W(2)

مساو لجملة الإعلان السابقة .

تحتجز أى جملة من جمل COMMON المذكورة عالية كتلة مشتركة مكونة من أماكن ذاكرة عددها (2 + 1 + 6 + 1) حيث يتم تحديد أماكن الذاكرة تبعاً لترتيب قائمة المتغيرات . بالتحديد ، وفي هذا المثال ، تسمى أماكن الذاكرة كما في شكل ١١ - ١ .

ونوضح هنا كيف تستخدم جملة COMMON وكيف تقسم مساحة التخزين المشتركة بواسطة هيكل البرنامج المعطى في شكل ١١ - ٢ . بمجرد التبسيط فإن البرامج الفرعية تستخدم نفس جملة COMMON وبمنتهى الدقة ، رغم أن ذلك غير إلزاي . انظر ملاحظة ١ على صفحة ٣١٩ . لاحظ أن البرنامج الرئيسى يعرف مساحة تخزين مشتركة تحتوى على 20 مكاناً من أماكن الذاكرة ويستعمل ثلاثة برامج فرعية وبرنامج دالة فرعياً .

البرنامج الرئيسي

```

COMMON A(10), B, X(2, 4), Q
.....
CALL XXX(S, I)
.....
CALL WWW(P1, P2, P3)
.....
T = TAX + ZZZ(U, V)
.....
CALL YYY
.....
END

```

البرنامج الفرعي رقم ١

```

SUBROUTINE XXX(F, K)
COMMON A(10), B, X(2, 4), Q
.....
RETURN
END

```

البرنامج الفرعي رقم ٢

```

SUBROUTINE YYY
COMMON A(10), B, X(2, 4), Q
.....
RETURN
END

```

البرنامج الفرعي رقم ٣

```

FUNCTION ZZZ(G, H)
COMMON A(10), B, X(2, 4), Q
.....
ZZZ = .....
RETURN
END

```

البرنامج الفرعي رقم ٤

```

SUBROUTINE WWW(F, G, H)
B = 0.0
.....
RETURN
END

```

شكل ١١ - ٢

X	
Y(1, 1)	
Y(2, 1)	
Y(1, 2)	
Y(2, 2)	
Y(1, 3)	
Y(2, 3)	
Z	
W(1)	
W(2)	

شكل ١١ - ١

تحتوي البرامج الفرعية XXX و YYY ودالة البرنامج الفرعي ZZZ على جملة COMMON ومن ثم ، سيكون له حق التوصل لمساحة التخزين المشتركة . وعموماً فكل برنامج من البرامج يستطيع أن يستخدم أو يغير أى قيمة مخزنة في هذه المساحة . من ناحية أخرى ، حيث أن البرنامج الفرعي WWW لا يوجد به جملة COMMON فبالتالى ، ليس له حق التوصل لمساحة التخزين المشتركة .

موجودة واحدة أخرى نريد أن نؤكدها . وهى أن الجملة :

$$B = 0.0$$

في البرنامج الفرعي WWW ليس لها تأثير على مساحة التخزين المشتركة حيث أن B خاصة فقط بالبرنامج الفرعي WWW ومع ذلك إذا ظهرت هذه الجملة في أى من البرامج الفرعية الأخرى . فسوف تتغير قيمة B في مساحة التخزين المشتركة وتصبح 0.0 بعد تنفيذها . علاوة على ذلك فستكون هذه القيمة B متاحة لأى برامج أخرى بعد ذلك .

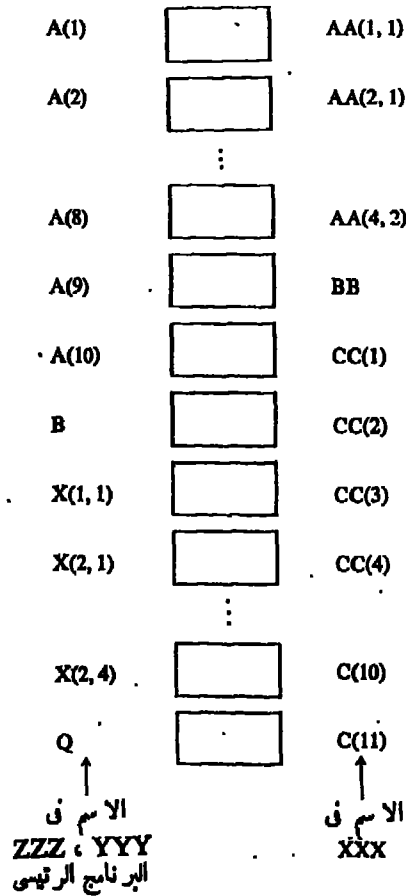
ملاحظة ١ : استخدام جملة COMMON غير المميزة (الخالية) يخلق مساحة تخزين مشتركة واحدة فقط . يمكن التوصل إليها ببرامج فرعية مختلفة . ومع ذلك فأماكن الذاكرة في هذه المساحة المشتركة يمكن أن تسمى بأسماء مختلفة في البرامج الفرعية المختلفة . فمثلا ، نفرض أن البرنامج الفرعي XXX السابق له الشكل التالي :

```
SUBROUTINE XXX(F, K)
COMMON AA(4, 2), BB, CC(11)
.....
RETURN
END
```

رغم أن جملة COMMON تشير إلى نفس مساحة التخزين المشتركة ، إلا أن أماكن الذاكرة في هذه المساحة المشتركة للتخزين تسمى بأسماء مختلفة في البرامج المختلفة كما هو موضح في الشكل ١١ - ٣ . وبذلك إذا ظهرت الجملة :

CC(2) = 5.5

في البرنامج الفرعي XXX وتم تنفيذها في البرنامج الفرعي ، لتغيرت قيمة B أيضاً في البرامج الفرعية YYY و ZZZ وفي البرنامج الرئيسي وأصبحت 5.5 .



شكل ١١ - ٣

ملاحظة ب : يمكن أن يكون عدد خلايا الذاكرة المرفقة في المساحة المشتركة COMMON في البرنامج الفرعي أصغر من المساحة المشتركة المعرفة في البرنامج الرئيسي . فثلا . مسموح بالتالي :

برنامج رئيسي COMMON X, A(5), Y
برنامج فرعي COMMON B(2, 2)

وعلاوة على ذلك ستقسم المتغيرات خلايا الذاكرة في المساحة المشتركة COMMON كما يلي :

الاسم في البرنامج الرئيسي الاسم في البرنامج الفرعي	X B(1, 1)	A(1) B(2, 1)	A(2) B(1, 2)	A(3) B(2, 2)	A(4)	A(5)	Y

لاحظ أن العناصر B(1,1), X في جملة COMMON تقسم نفس المكان . ومن ناحية أخرى ، لا يمكن أن تتجاوز عدد خلايا الذاكرة في المساحة المشتركة COMMON الموجودة في البرنامج الفرعي تلك الموجودة في البرنامج الرئيسي . فثلا غير مسموح بما يلي :

برنامج رئيسي COMMON C(2, 3)
برنامج فرعي COMMON D(8)

ملاحظة ج : يمكن استخدام عدة جملة COMMON غير مميزة في برنامج واحد . ومع ذلك فإنه يكافئ كتابة جملة COMMON واحدة طويلة . فثلا :

COMMON A(10), T, C(2) تكافئ
COMMON A(10), T
COMMON C(2)

ويتم حجز 13 مكاناً من أماكن الذاكرة للمساحة المشتركة COMMON.

١١ - ٩ جملة COMMON المميزة

تخلق جملة COMMON غير المميزة مساحة تخزين مشتركة واحدة فقط . وباستخدام جملة COMMON المميزة يمكن أن نعرف عدة مساحات تخزين مشتركة . يستدل على ذلك من اسم جملة COMMON فلكل من هذه المساحات المشتركة لتخزين عنوان (اسم) . وهذه الجملة غير منفذة أيضاً مثل جملة COMMON غير المميزة ، ، فيما يلي نموذج لجملة COMMON المميزة :

COMMON /A/ TAX(10), X, INF(2, 4)

لا تملن هذه الجملة أن TAX و INF أسماء مجموعات مترابطة فقط ولكنها تعرف أيضاً مساحة تخزين مشتركة مسماة (لها عنوان) باسم A وبها 19 مكاناً من أماكن التخزين كما هو موضح في شكل ١١ - ٤ . لاحظ أن جملة COMMON المميزة تبدأ بكلمة COMMON ويتبها عنوان (اسم) الكتلة المشتركة محاطاً بشرطات مائلة . (ويتكون العنوان ، مثل أي اسم متغير آخر من عدد يصل إلى ستة حروف أبجدية مع ضرورة كون الحرف الأول أبجدياً) يلي بعد ذلك قائمة المتغيرات وأسماء المجموعات المترابطة المعطاة لأمكن الذاكرة في هذه المساحة المشتركة للتخزين .

يمكن أن نعرف عدة مساحات تخزين مشتركة مميزة في جملة COMMON واحدة . فثلا :

COMMON /A/ TAX(10), X, INF(2, 4)/B/POP(100)

تعرف مساحتان مشتركتان . أحدهما باسم A والأخرى باسم B . ونذكر القارئ أن المجموعات المترابطة يمكن أن تعرف إما في جملة DIMENSION أو جملة COMMON ولكن ليس في كليهما . وبذلك ، أي من الترتيبات الآتية يكافئ الجملة السابقة :

A	
TAX(1)	
⋮	
TAX(10)	
X	
INF(1, 1)	
INF(2, 1)	
⋮	
INF(2, 4)	

شكل ١١ - ٤

DIMENSION TAX(10), INF(2, 4), POP(100)
COMMON /A/TAX, X, INF /B/POP

أو

DIMENSION TAX(10)
COMMON /A/TAX, X, INF(2, 4)
COMMON/B/POP(100)

وأنا لتؤكد عدم وجود فصلة تفصل بين المساحات المشتركة المميزة في جملة COMMON الواحدة (فبلا لا توجد فصلة تسبق /B/ في أى من الجمل السابقة) .

الميزة الأساسية للمساحات المشتركة المسماة (المميزة) هي إعطاء الحق لبرامج فرعية مختلفة في التوصل لمساحات مشتركة مختلفة . وأنه لغاية في الأهمية أن نتذكر أن اسم الكتلة المشتركة هو الذى يتداول من برنامج فرعى إلى آخر . أما أسماء خلايا الذاكرة بداخل كتلة مشتركة معينة فهي محلية لهذا البرنامج الفرعى ، ومن ثم ، يمكن أن تكون مختلفة . كما تم مناقشتها سابقاً في حالة الكتلة المشتركة غير المميزة . ونوضح ذلك بالمثال التالى .

مثال ١١ - ٢

ادرس هيكل البرنامج في شكل ١١ - ٥ . لاحظ أنه تم تعريف مساحتين مشتركين باسم AX و AY البرنامج الفرعى SUB1 له حق التوصل للمساحة المشتركة AY التى تحتوى على ثمانية أماكن للذاكرة وأسماء هذه الأماكن في البرنامج الفرعى هي نفس الأسماء في البرنامج الأساسى ، أى X(1,1) و X(2,1) و ... و X(2,4) و يقسم البرنامج الفرعى SUB2 مع البرنامج الرئيسى للمساحة المشتركة المسماة AX التى بها 11 مكاناً للذاكرة . تسمى هذه الأماكن A(1) و ... و A(10) في البرنامج الرئيسى ولكن تسمى T(1,1) و ... و T(2,3) و S(1) و ... و S(5) في البرنامج الفرعى كما هو موضح في الشكل ١١ - ٦ .

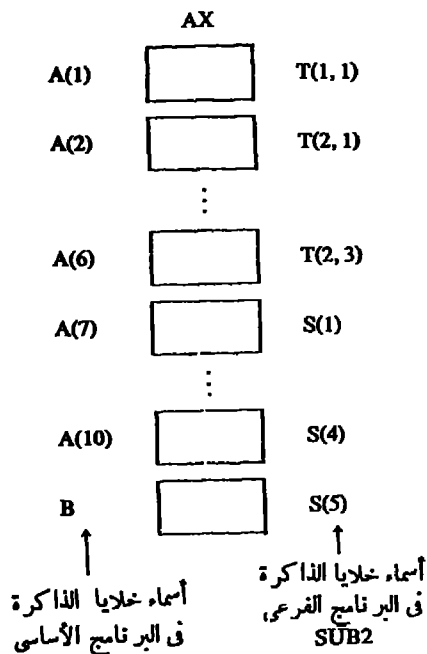
ونختتم بلحوظات قليلة أخرى حول جمل COMMON المميزة :

١ - إذا تشابه اسم مساحتين مشتركتين في برنامج فرعى ، كما يلي :

```
COMMON /A/T(2, 4)/B/X, Y, Z /A/Q(10)
COMMON /B/W(10)
```

فتكون النتيجة تجميعها ، أى يمكننا كتابة الجمل السابقة كما يلي :

```
COMMON /A/T(2, 4), Q(10) /B/X, Y, Z, W(10)
```



شكل ١١ - ٦

```
البرنامج الأساسى
DIMENSION X(2, 4)
COMMON/AX/A(10),B /AY/X
.....
CALL SUB1
CALL SUB2(C, D)
.....
END
```

```
البرنامج الفرعى رقم ١
SUBROUTINE SUB1
COMMON /AY/X(2, 4)
.....
RETURN
END
```

```
البرنامج الفرعى رقم ٢
SUBROUTINE SUB2(E, F)
COMMON/AX/T(2, 3), S(5)
.....
RETURN
END
```

شكل ١١ - ٥

٢ - يجب أن يكون العدد الكلى لأماكن الذاكرة في مساحة مشتركة مميزة هو نفس العدد في كل البرامج الفرعية التى لها حق التوصل لهذه المساحة . لاحظ أن هذا يختلف عما سبق ذكره عن المساحة غير المميزة .

٣ - يمكن أن تعرف المساحات المشتركة المميزة وغير المميزة في نفس البرنامج باستخدام نفس جملة COMMON فثلا تخلق الجملة التالية :

```
COMMON TAX(100), A, B /Q/X(25, 10)
```

مساحة مشتركة غير مميزة بها عدد 102 خلية ذاكرة ومساحة مشتركة مميزة باسم Q بها عدد 250 خلية ذاكرة . ويمكن أن نستخدم الجملة التالية لتفى بنفس الغرض :

```
COMMON /Q/X(25, 10)//TAX(100), A, B
```

لاحظ عدم وجود اسم بين الشرطيات الماثلة بالنسبة لمساحة التخزين الثانية . ومن ثم نخلق مساحة مشتركة غير مميزة كما سبق ذكره . هذا هو السبب في أن جملة COMMON غير المميزة تسمى أيضاً جملة COMMON الحالية .

يمكن أن يكون استخدام جملة COMMON المميزة وغير المميزة بصورة عامة معقداً للغاية ويؤدى للارتباك، لذلك فإننا نعتبر أن أى مناقشة لهذا الموضوع أكثر من ذلك تكون خارج نطاق هذا الكتاب .

١١ - ١٠ جملة EQUIVALENCE

افرض أن المبرمج أدرك بعد كتابة برنامجاً طويلاً وبعد تثقيبه ، أنه استخدم أسماء مختلفة ، وليكن MAX و ARG و JBIG لنفس الكمية ، فيمكن أن يضيف المبرمج ببساطة جملة EQUIVALENCE في بداية البرنامج بدلا من تغيير كل الأسماء وإعادة تثقيب البطاقات .

EQUIVALENCE (MAX, LARGE, JBIG)

تغير هذه البطاقة الحاسب أن الأسماء MAX و LARGE و JBIG تمثل كلها نفس أماكن الذاكرة .

لاحظ أن جملة EQUIVALENCE تبدأ بكلمة EQUIVALENCE وتنتج بأقواس تحيط بتلك المتغيرات التي تشير إلى نفس أماكن الذاكرة .

يمكن أن تستخدم جملة EQUIVALENCE أيضاً لتوفير أماكن الذاكرة وذلك بتخصيص أسماء مختلفة إلى نفس أماكن الذاكرة ، بشرط أن تستخدم المتغيرات المختلفة في أجزاء مختلفة من البرنامج . فثلا ، إذا أردنا كتابة برنامج لحساب المتوسط AVE والوسيط XMED والقيمة الكبرى BIG لعدد 100 رقم تم طباعتها . افرض أننا لن نحتاج المتغيرات AVE و XMED و BIG في نفس الوقت فيمكن أن نوفر مساحة التخزين بتخصيص المتغيرات الثلاثة إلى نفس أماكن الذاكرة باستخدام جملة EQUIVALENCE

EQUIVALENCE(AVE, XMED, BIG)

وكما هو متبع ، فيمكن أن نستخدم جملة EQUIVALENCE واحدة لتحديد عدة مجموعات متساوية . فثلا . :

EQUIVALENCE(A, X, Z), (NEXT, LAST), (C, D, E, F)

تغير الحاسب أن A و X و Z سوف تشترك في نفس مكان التخزين ، وسوف تشترك NEXT و LAST في نفس مكان التخزين وأن C و D و E و F سوف تشترك في نفس مكان التخزين .

يحدث التطبيق الرئيسى لجملة EQUIVALENCE مع المجموعات المتراسة حيث يمكن توفير آلاف من أماكن التخزين . ومع ذلك يمكن أن تكون الطريقة التي تحتل بها المجموعات المتراسة نفس المساحة طريقة خادعة ، وبذلك فجملة EQUIVALENCE التي تشمل مجموعات متراسة ليست ببساطة الحمل الخاصة بالمتغيرات بدون دليل والتي تنا بمناقشتها مسبقاً .

. افرض مثلاً أن A و B و C مجموعات متراسة في برنامج وقد تم تعريفها بواسطة :

DIMENSION A(50, 100), B(25, 50), C(3750)

افرض أننا عرفنا بعد ذلك أن المجموعة المتراسة A مطلوبة في بداية البرنامج فقط ولا تستخدم في حسابات المجموعة المتراسة B و C نستطيع أن نوفر أماكن الذاكرة بتخزين المجموعات المتراسة B و C في أماكن الذاكرة المهجوزة لـ A وبالتحديد تشير إلى أول عنصر من A وأول عنصر من B سيكون لهما نفس أماكن الذاكرة بواسطة :

EQUIVALENCE(A(1, 1), B(1, 1))

حيث أن عناصر المجموعات المتراسة متصلة فإن هذه الجملة تخبر الحاسب أيضاً أن أول 1250 مكان ذاكرة من A ستحتل نفس الأماكن مثل 1250 مكان من B (هل تستطيع أن تحدد التناظر ؟) يمكن أن تستخدم 3750 مكاناً الباقية من A للمجموعة المتراسة C بواسطة الجملة التالية :

EQUIVALENCE(A(1, 26), C(1))

يمكن أن تدمج هاتين الجملتين في جملة واحدة كما يلي :

EQUIVALENCE(A(1, 1), B(1, 1)), (A(1, 26), C(1)).

وتتداخل الآن مساحات التخزين ويتم حجز 5000 مكان من الذاكرة فقط بدلا من 10,000 مكان . الشيء القاطع الذي يجب أن نتذكره حول المجموعات المتراسة هو أن :

١ - عناصر مجموعة المتراسة تعطى دائماً جنباً إلى جنب في ترتيب خطي، وهذا لا يمكن تغييره بأي جملة **EQUIVALENCE** أخرى .

٢ - يمكن أن تظهر عناصر المجموعة المتراسة فقط في جملة **EQUIVALENCE** وليس اسم المجموعة المتراسة نفسها .

ملاحظة : تذكر أن المجموعات المتراسة تخزن خطياً في الذاكرة ورغم أن المجموعات المتراسة يمكن أن تكون متعددة الأبعاد . تسمح بعض المترجات باستخدام المكان الخطي لعناصر أي مجموعة متراسة في جملة **EQUIVALENCE**

مثال : ، تحتل A(1, 1) المكان الأول في المجموعة المتراسة A السابقة وتحتل A(1, 26) المكان 1251 من A . وتشغل أيضاً B(1, 1) المكان الأول في B وبالتالي فيمكن استبدال الجملة :

EQUIVALENCE(A(1, 1), B(1, 1)), (A(1, 26), C(1))

بواسطة الجملة :

EQUIVALENCE(A(1), B(1)), (A(1251), C(1))

والميزة في استخدام المكان الخطي هو سهولة حساب المكان .

في أي من الأشكال سابقة يجب أن نذكر أن الأدلة لا بد وأن تكون قياً ثابتة إذ أن جملة **EQUIVALENCE** غير منفذة ، وتحجز أماكن التخزين أثناء الترجمة .

نعم بعد ذلك أمثلة أخرى تشمل مجموعات متراسة مع جمل **EQUALENCE** .

مثال ١١ - ٣

(١) ادرس الجمل التالية :

DIMENSION A(2, 3)
EQUIVALENCE(A(2), X, Y), (A(5), C)

تشارك المتغيرات أماكن التخزين كما يلى :

A(1, 1)	A(2, 1) X Y	A(1, 2)	A(2, 2)	A(1, 3) C	A(2, 3)

أى ، يخصص لكل من A(2,1) و X و Y نفس أماكن التخزين ، ويخصص لكل من A(1,3) و C نفس أماكن التخزين .

(ب) افرض إمكانية استخدام المكان الحطى لمجموعة متراسة . ادرس الجمل التالية :

DIMENSION A(2, 3), B(2, 2), C(5)
EQUIVALENCE(A(1), B(1), C(1))

تشارك المجموعات المتراسة مساحة التخزين كما يلى :

A(1, 1) B(1, 1) C(1)	A(2, 1) B(2, 1) C(2)	A(1, 2) B(1, 2) C(3)	A(2, 2) B(2, 2) C(4)	A(1, 3) C(5)	A(2, 3)

لاحظ أن C(4) أجبرت على مشاركة المكان مع A(2,2) و B(2,2) حيث تم تخصيص أول عنصر فى كل مجموعة متراسة إلى نفس المكان .

(ج) ادرس الجمل التالية :

DIMENSION A(2, 3), C(7)
EQUIVALENCE(A(3), C(1))

فسوف تظهر المجموعات المتراسة فى مساحة التخزين كما يلى :

A(1, 1)	A(2, 1)	A(1, 2) C(1)	A(2, 2) C(2)	A(1, 3) C(3)	A(2, 3) C(4)	C(5)	C(6)	C(7)

لاحظ أن المجموعة المتراسة C ترحل من الجانب الأيمن .

نستطيع أن نستخدم جملة EQUIVALENCE مع جملة COMMON ومع ذلك ، فيمكن لجملة EQUIVALENCE أن تزيد من طول المخزن المشترك فقط ، ولكن لا يمكنها أن تغير الأماكن الأصلية للمتغيرات . نوضح ذلك بأثلة .

مثال ١١ - ٤

(١) ادرس الجمل التالية :

DIMENSION A(4), C(2), D(8)
COMMON A, B, C
EQUIVALENCE(A(3), D(1))

ستظهر مساحة التخزين المشتركة كما يلي :

```
A(1) A(2) A(3) A(4) B C(1) C(2)
      D(1) D(2) D(3) D(4) D(5) D(6) D(7) D(8)
```

(لم نرمم صناديق مع أسماء المتغيرات كما فعلنا في مثال ١١ - ٣ وذلك لسهولة الترميز) تخلق جملة COMMON الأصلية سبعة أماكن في المخزن المشترك . تزيد جملة EQUIVALENCE مساحة المخزن المشترك إلى 10 أماكن .

(ب) حمل FORTRAN التالية غير مسموح بها :

```
DIMENSION A(4), C(2), D(8)
COMMON A, B, C
EQUIVALENCE(A(1), D(3))
```

وللسبب السابقه يجب أن يبدو اقتسام المخزن المشترك كالتالى :

```
A(1) A(2) A(3) A(4) B C(1) C(2)
D(1) D(2) D(3) D(4) D(5) D(6) D(7) D(8)
```

تمثل (١) هنا المركز الثالث ، ولكن تيمناً لجملة COMMON يجب أن تحتل A(1) المركز الأول حيث أن الفورتران لا يسمح لنا بتغيير الأماكن الأصلية للمتغيرات في المخزن المشترك ، لذلك فإن هذه الجملة غير مسموح بها .

يمكن أن تكون الاستخدامات الأخرى لجملة EQUIVALENCE وبخاصة التي تشمل جملة COMMON معقدة جداً وتعتمد على الآلة ؛ من ثم ، فتعتبر أى مناقشة أخرى خارج نطاق هذا الكتاب . وعلاوة على ذلك ، نظراً لأنه من الممكن الاستغناء عن جملة EQUIVALENCE كلية فإننا ننصح المبرمج المبتدئ أن يتجنب استخدام جملة EQUIVALENCE كلياً . يمكن ذلك .

١١ - كتلة البيانات

لا تسمح بعض المترجمات باستخدام جملة DATA لإعطاء قيم ابتدائية لمتغيرات في مساحات COMMON المميزة ، ولا يسمح أب . بإعطاء قيم ابتدائية لأى مساحات COMMON إلا أنه يمكن أن نستخدم برنامجاً فرعياً يسمى BLOCK DATA لهذا الغرض .

يمكن أن يتكون البرنامج الفرعى BLOCK DATA من الجملة غير المنفذة التالية فقط :

```
DATA
COMMON
DIMENSION
EQUIVALENCE
```

وأي إعلانات عن النوع . وسوف نوضح كيفية عمل مثل هذا البرنامج الفرعى بمثال

افرض أن لدينا الآتى في برنامج رئيسى :

```
DIMENSION X(20), Y(15)
COMMON /A/X, U, V /B/Y
      :
      :
      :
END
```

افرض أننا نريد إعطاء قيمة ابتدائية 0.0 لكل عنصر من X و Y وحيث أن X و Y فى مساحات مشتركة COMMON مميزة فإن استخدام جملة DATA غير مسموح بها . ولكننا نستطيع أن نستخدم جملة تخصيص لإعطاء القيم الابتدائية التى نحتاجها وقت التنفيذ . من ناحية أخرى يمكن أن نستخدم البرنامج الفرعى BOLCK DATA الآتى :

```
BLOCK DATA
DIMENSION X(20), Y(15)
COMMON /A/X, U, V /B/Y
DATA X/20*0.0/, Y/15*0.0/
END
```

للمنص فيما يلى استخدام البرنامج الفرعى BLOCK DATA :

- ١ - يبدأ البرنامج الفرعى دائماً بالجملة BLOCK DATA .
- ٢ - يجب أن نذكر كل المتغيرات الموجودة فى المساحة COMMON فى البرنامج الفرعى برغم أننا نريد أن نعطى قيمة ابتدائية لبعض المتغيرات فقط .
- ٣ - يمكن أن يستعمل برنامجاً فرعياً BLOCK DATA واحداً فقط لإعطاء قيم ابتدائية للمتغيرات فى أى مساحة مشتركة COMMON واحدة .

١١ - ١٢ خارجى EXTERNAL

لذاكر أن الاتصال الوحيد بين البرنامج الدامى وأى برنامج فرعى (إلى جانب COMMON و EQUIVALENCE) هو عن طريق قائمة الخلاصات . يكون مستخدماً فى بعض الأحيان أن نستخدم اسم برنامج فرعى كخلاصة حقيقية فى جملة الاستدعاء لبرنامج فرعى آخر . هيكل البرنامج التالى يوضح مثل هذا الموقف :

```
البرنامج الأساسى
.....
EXTERNAL TOTAL
X = BAL(ASSET, PAY, TOTAL)
.....
} برنامج فرعى } END
FUNCTION BAL(A, B, C)
T = C(A)
.....
RETURN
} برنامج فرعى } END
FUNCTION TOTAL(Z)
.....
RETURN
END
```

كما هو مشار إليه بأحد الأسهم ، فاسم البرنامج الفرعى TOTAL يتم نقله إلى البرنامج الفرعى BAL باستخدام المعامل C . لذلك ، تحسب C(A) فى BAL باستخدام البرنامج الفرعى TOTAL كما هو مشار إليه بالسهم الآخر .

ولذلك فيكون إعلان خاص ، لن يتمكن المترجم من التفريق بين اسم البرنامج الفرعى واسم متغير بسيط يحمل قيمة . ولتجنب التثريب يستخدم إعلان EXTERNAL فى البرنامج الدامى لتشير أن اسم البرنامج الفرعى المستخدم كخلاصة هو خارجى بالنسبة لهذا البرنامج .

مسائل

جمل IMPLICIT وجمل النوع TYPE

١١ - ١ اكتشاف الأخطاء ، إن وجدت ، في كل جزء برنامج مما يأتى :

INTEGER, RATE, TIME, DISTANCE	(١)
DIMENSION R(4), S(5, 3)	(ب)
DOUBLE PRECISION A(7), B, R(8), S,	(ج)
IMPLICIT, COMPLEX (A - D, F, G) LOGICAL(C, X - Z)	(د)
IMPLICIT REAL M, N, COMPLEX P - R	(هـ)
DOUBLE PRECISION NUMBER(4, 7) X, Y, TEST(8, 15) L, M	(١)

١١ - ٢ ادرس جزء البرننج التالى :

IMPLICIT COMPLEX(H - K) LOGICAL(P, R)
REAL COEF, KAPPA, LAMBDA
INTEGER RATE, TIME

حدد نوع كل متغير مما يلى :

ROOM (ى)	LAST (ز)	KOUNT (د)	ALPHA (١)
TIME (ك)	LAMBDA (ح)	PARTY (هـ)	INDEX (ب)
TARGET (ل)	RATE (ط)	QUEUE (و)	KAPPA (ج)

دقة متضاعفة

١١ - ٣ حدد ما إذا كان كل متغير يحتل ثابتاً حقيقياً (دقة مفردة) أو دقة متضاعفة (مع فرض سبع خانوات مبنوية للثوابت الحقيقية) :

123.123123 + 0.E0 (ز)	12345678 (د)	1.45D + 03 (١)
5.55E - 26 + 58.4 (ح)	23.4 + 0.D0 (هـ)	5.9321E - 7 (ب)
6.7 + 4.5D1 - 7.8E3 (ط)	111.222333E - (و)	3456.765432 (ج)

١١ - ٤ حدد المخرج لكل جزء من البرامج التالية :

DOUBLE PRECISION X, Y, Z (ب)	DOUBLE PRECISION DA (١)
X = 70/9	A = 0.555666777888999
Y = 70.0/9.0	DA = 0 55566677788899
Z = 70.0D0/9.0D0	WRITE(6, 10) A, A
WRITE(6, 30) X, Y, Z	10 FORMAT(1X, E25.3, 5X, E25.9)
30 FORMAT(1X, D20.12)	WRITE(6, 20) DA, DA
	20 FORMAT(1X, D25.3, 5X, E25.9)
	STOP
	END

١١ - ٥ افرض أن $A = 9999999.0$, $B = 125.0$, and $C = 9999998.0$ ولاحظ أن

$$X = A + B - C = 126.0$$

(حيث يمكن الحصول عليها باستخدام حساب الدقة المتضاعفة) .

(١) احسب قيمة X باستخدام حساب الدقة المفردة (مع فرض سبع خانات معنوية)

(ب) أوجد الخطأ وكذلك نسبة الخطأ المئوية .

١١-٦ اكتب البرنامج الذى يحسب مضروب n أى $(n!)$ لـ $n = 1, 2, \dots, 15$ (يجب أن يستخدم حساب الدقة المتضاعفة حيث أن كلا من الحساب الصحيح وحساب الدقة المفردة ليس بالدقة الكافية) .

١١-٧ تذكر أنه يمكن حساب مساحة المثلث ذى الأطوال a و b و c بواسطة :

$$\text{AREA} = \sqrt{s(s-a)(s-b)(s-c)}$$

حيث :

$$s = (a + b + c)/2$$

اكتب البرنامج الذى يقرأ قيم a و b و c ويحسب ويطبغ المساحة AREA باستخدام حساب الدقة المتضاعفة .

أعداد مركبة

١١-٨ اجمل $z = 2 - 3i$ و $w = 4 + 5i$ أوجد ما يأتى جبرياً :

(١) $z + w$ (ب) zw (ج) z/w (د) \bar{z} and \bar{w} (هـ) $|z|$ and $|w|$

تحقق من النتائج على الحاسب باستخدام برنامج فورتران :

١٠-٩ بفرض أن X و Y و Z متغيرات مركبة و A و B و C متغيرات حقيقية ، اكتشف الأخطاء إن وجدت فى كل جملة مما يأتى :

$$Z = 3.6 + (5.0, 7.6) \quad (\text{ج})$$

$$X = (3, 4.5) \quad (\text{أ})$$

$$A = \text{COMPLEX}(B, C) \quad (\text{د})$$

$$Y = (A, B + C) \quad (\text{ب})$$

١٠-١٠ افرض أنه تم تثقيب أول بطاقتين للبيانات كالتالى :

	1	2					
	1234567890	1234567890	12345				
البطاقة الأولى	1.1	2.2	3.3	4.4			
البطاقة الثانية	5.5	6.6	7.7	8.8			

أوجد الخرج لكل جزء برنامج ما يلى :

```

COMPLEX A, B, C (ب)
READ(5, 50) A, B
50 FORMAT(3F5.1)
C = A*B
WRITE(6, 60) C
60 FORMAT(1X, 'REAL PART IS', 7X, F10.2/1X,
1 'IMAGINARY PART IS', 2X, F10.2)

```

```

COMPLEX A, B, C (١)
READ(5, 10) A, B
10 FORMAT(2F5.1)
C = A*B
WRITE(6, 20) C
20 FORMAT(1X, F10.2)

```

١١-١١ اكتب برنامج يقرأ ثلاثة أرقام مركبة ويطبع الأرقام الثلاثة على سطر ثم يطبع مجموعها وحاصل ضربها على سطر آخر .

١١-١٢ اكتب برنامجاً يقرأ الأرقام الحقيقية a و b و c ويحل معادلة الدرجة الثانية التالية :

$$ax^2 + bx + c = 0$$

باستخدام الصيغة الرياضية :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(لاحظ أن الجذور تكون مركبة إذا كان مميز المعادلة $D = b^2 - 4ac$ أقل من صفر) .

جمل GO TO و ASSIGN والمداخل والعودة المتعددة

١١-١٣ اكتشف الأخطاء ، إن وجدت ، في كل جملة من جمل الفورتران التالية :

GO TO NUMBER (5, 78, 6, -43) (ج) ASSIGN +45 TO NUMBER (أ)
GO TO 200, (75, 25, 200, 50) (د) ASSIGN N TO NUMBER (ب)

١١-١٤ اكتشف الأخطاء ، إن وجدت ، في كل جزء من برامج الفورتران التالية :

ASSIGN 25 TO NEXT (ب) ASSIGN 25 TO NEXT (أ)
NEXT = 50
GO TO NEXT, (25, 40, 50, 70) GO TO NEXT, (50, 75, 300)

١١-١٥ اكتشف الأخطاء إن وجدت ، في هيكل :
: نامج التالى :

البرنامج الرئيسى	البرنامج الفرعى
.....	ROUTINE BIG(A, B, C)
CALL BIG(X, Y, Z)
.....	E MIDDLE(X, Y, Z)
CALL MIDDLE(P, A + B)
.....	EN RATE(C, D, K)
CALL RATE(X + Y, J, N + 4)
.....	END
STOP	
END	

١١-١٦ ادرس جملة ALL رتية وكذا جملة SUBROUTINE المناظرة :

CALL UPDATE(A, 88, B, 36, C, N, 44)
SUBROUTINE UPDATE(X, *, Y, *, Z, NUMBER, *)

أوجد النقطة التي ينتقل إليها التحكم إذا قابلنا الجملة التالية في البرنامج الفرعى UPDATE :

RETURN 1 (أ) RETURN 2 (ب) RETURN 3 (ج) RETURN (د)

EQUIVALENCE و COMMON

١٧-١١ اكتشف الأخطاء ، إن وجدت ، فى كل جملة من جمل فورتران التالية :

COMMON (X, Y, Z)	(١)
EQUIVALENCE X, Y, Z	(ب)
COMMON X, Y(2, 4), Z(8), W.	(ج)
EQUIVALENCE (X, Y) (A(1), B(4), Z)	(د)
COMMON /A/ NUM(5), X, /B/ INDEX(4, 5)	(هـ)

١٨-١١ ارسم شكل المخزن الذى ينتج من هيكل البرنامج التالى :

برنامج رئيسى	برنامج فرعى
COMMON Y(4), A, B, C(2)	SUBROUTINE NNN(P1, P2, P3)
.....	COMMON D, X(2), E(2), F(2)
CALL NNN(P, Q, R)
.....	END
END	

١٩-١١ حدد شكل المخزن الذى ينتج من أجزاء البرنامج التالية :

DIMENSION X(2, 3), Y(3, 2) (ب)	DIMENSION A(5), B(4), C(7) (١)
EQUIVALENCE(X(3), Y(1))	EQUIVALENCE (A(1), B(1), C(2))

٢٠-١١ افرض أن A مصفوفة أى مجموعة متراسة (3 × 3) ونريد أن نجعل DIAG مجموعة متراسة خطية بها ثلاثة عناصر تحدد مداخل القطر لـ A ناقش الخطأ فى جزء البرنامج التالى :

DIMENSION A(3, 3), DIAG(3)
EQUIVALENCE (A(1), DIAG(1)), (A(5), DIAG(2)), (A(9), DIAG(3))

اجابات المسائل المختارة

- ١٠-١ (١) يجب ألا تكون هناك فصلة بعد INTEGER .
 (ب) تم تعريف R مرتين بأحجام مختلفة . يجب ألا تكون هناك فصلة بعد S .
 (ج) يجب ألا تكون هناك فصلة بعد IMPLICIT يجب أن تكون هناك فصلة قبل LOGICAL .
 (د) COMPLEX (P—R) و IMPLICIT REAL (M, N)
 (هـ) يجب أن تكون هناك فصلة قبل X وفصل قبل L .

٢-١١

- (١) حقيقى (ب) مركب (ج) حقيقى (د) مركب (هـ) منطوقى (و) حقيقى
 (ز) صحيح (ح) حقيقى (ط) صحيح (ى) منطوقى (ك) صحيح (ل) حقيقى

٢-١١

(أ) مضاعف (ب) حقيقي (ج) مضاعف (د) لا هذا ولا ذلك ، حيث أنه صحيح
(هـ) مضاعف (و) مضاعف (ز) مضاعف (ح) حقيقي (ط) مضاعف

0.700000000000D 01 (ب) 0.556E 00 0.555666700E 00 (أ) ٤-١١
0.777777700000D 01 0.556D 00 0.555666778D 00
0.77777777778D 01

٥-١١ (أ) 1.2.0 (ب) خطأ = 4 ، وهي 3 في المائة .

٧-١١ يجب أن تدعى كل المتغيرات DOUBLE PRECISION ويجب أن تستخدم دالة الجذر التربيعي ذات الدقة المتضاعفة
: DSQRT

```
DOUBLE PRECISION A, B, C, S, AREA
READ(5, 10) A, B, C
10 FORMAT(3D20.10)
S = (A + B + C)/2.0
AREA = DSQRT(S*(S - A)*(S - B)*(S - C))
WRITE(6, 20) AREA
20 FORMAT(1X, 'AREA IS', 3X, D20.12)
STOP
END
```

٨-١١ (أ) $6 + 2i$ (ب) $23 - 2i$ (ج) $-7/41 - 22i/41$ (د) $2 + 3i, 4 - 5i$ (هـ) $\sqrt{13}, \sqrt{41}$

٩-١١ (أ) يجب أن تكتب 3 في الصورة 3.0

(ب) CMPLX (A, B + C) وليس (A, B + C)

(ج) صواب .

(د) CMPLX (B, C) وليس COMPLEX (B, C)

REAL PART IS (ب) -8.47 (أ) ١٠-١١
IMAGINARY PART IS 13.31 19.36

١٣-١١ (أ) يجب أن تكون 43 بدون إشارة لكونها رقم جملة .

(ب) يجب أن تكون N رقم جملة حقيقياً ، وليس متغيراً صحيحاً .

(ج) يجب أن تكون هناك فصلة بعد NUMBER . 43 - غير مسموح بها أيضاً .

(د) يجب أن تكون أول 200 متغير صحيح ، وليس ثابتاً صحيحاً .

١٤-١١ (١) يجب أن تظهر القيمة 25 لـ NEXT في القائمة التالية لـ NEXT في جملة GO TO

(ب) يمكن تغيير قيمة NEXT بواسطة جملة ASSIGN أخرى فقط ، مثلا :

ASSIGN 30 TO NEXT

لا يمكن تغييره بواسطة جملة تخصيص حسابية .

١٥-١١ (١) MIDDLE في البرنامج الرئيسي لما خلاصتان فقط ولكن MIDDLE لما ثلاثة معاملات في البرنامج الفرعي . يجب أن تكون الخلاصة الثانية J في RATE حقيقية حيث أنها تناظر المعامل الحقيقي D .

١٦-١١ (١) إلى الجملة رقم 88 في البرنامج الداعي

(ب) إلى الجملة رقم 36 في البرنامج الداعي

(ج) إلى الجملة رقم 44 في البرنامج الداعي

(د) إلى أول جملة منفذة تل جملة الاستدعاء CALL .

١٧-١١ (١) يجب ألا تكون المتغيرات محاطة بأقواس .

(ب) يجب أن تكون المتغيرات محاطة بأقواس .

(ج) يجب ألا تكون هناك فصلة بعد W .

(د) يجب أن تكون هناك فصلة بعد (X, Y) .

(هـ) يجب ألا تكون هناك فصلة بعد X .

١٨-١١

Y(1) Y(2) Y(3) Y(4) A B C(1) C(2) : الاسم في البرنامج الرئيسي

D X(1) X(2) E(1) E(2) F(1) F(2) : الاسم في البرنامج الفرعي

١٩-١١

A(1) A(2) A(3) A(4) A(5)

B(1) B(2) B(3) B(4)

C(1) C(2) C(3) C(4) C(5) C(6) C(7)

(١)

X(1, 1) X(2, 1) X(1, 2) X(2, 2) X(1, 3) X(2, 3)

Y(1, 1) Y(2, 1) Y(3, 1) Y(1, 2) Y(2, 2) Y(3, 2)

(ب)

٢٠-١١ تشير المتساويات الثلاثة إلى أن DIAG (1) يجب أن تقسم مكان التخزين مع A(1, 1) و DIAG(2) مع A(2, 2) و DIAG(3) مع A(3, 3) كما يلي :

A(1, 1) A(2, 1) A(3, 1) A(1, 2) A(2, 2) A(3, 2) A(1, 3) A(2, 3) A(3, 3)

DIAG(1) DIAG(2) DIAG(3)

إلا أن ما سبق يتفصّل الشرط في أن عناصر المجموعة المترابطة يجب أن تظهر جنباً إلى جنب في المخزن ، أي ، أننا لا نستطيع أن نفصل عناصر DIAG

الفصل الثاني عشر

الفورتران الهيكلي

قد يحتاج الترق في عملية البرمجة لأكثر من عدة سنوات من التدريب والخبرة . وفي بعض الأحيان يصاب المبتدئون بالارتباك والأحباط عندما يتضح لهم أن برامجهم مجرد محسنة من الجمل بدون أي تنظيم أو منطق واضح .

يمكن أن نعزى منبع الارتباك والحلأ الأساسي في البرمجة بالفورتران إلى : (١) هياكل التحكم المربكة في لغة الفورتران نفسها ، و(٢) الاستخدام غير المنظم لبعض هياكل الفورتران ، وبالأخص جمل GO TO .

وعبر السنوات برزت عدة قواعد لممارسة جيدة للبرمجة أدت إلى فكرة البرمجة الهيكلية . وتبعاً لذلك فقد ضمنت كثير من التركيبات الهيكلية الدقيقة بداخل عدة مشغلات للفورتران . وقد ساعدت تركيبات الفورتران الهيكلية هذه في تخفيف ارتباك المبتدئين وتقليل نسبة الأخطاء ، فالبرامج المكتوبة بهذه الطريقة أكثر قابلية للقراءة ويمكن فهمها وتعديلها بسهولة أكثر .

يناقش هذا الفصل بعضاً من هياكل التحكم الشهيرة واستخدامها وعلاقتها بالفورتران غير الهيكلية . في الحقيقة فالقراء الذين لهم حق استخدام مشغلات الفورتران الهيكلية يمكنهم دراسة IF الهيكلية بعد الفصل الرابع وحلقة DO الهيكلية بعد الفصل الخامس .

١٢ - ! IF الهيكلية

هياكل IF في الفورتران الهيكلية لها تنوعات عديدة . ومع ذلك فهي تبدأ دائماً بكلمة جملة IF التي لها الشكل التالي :

IF(logexp) THEN

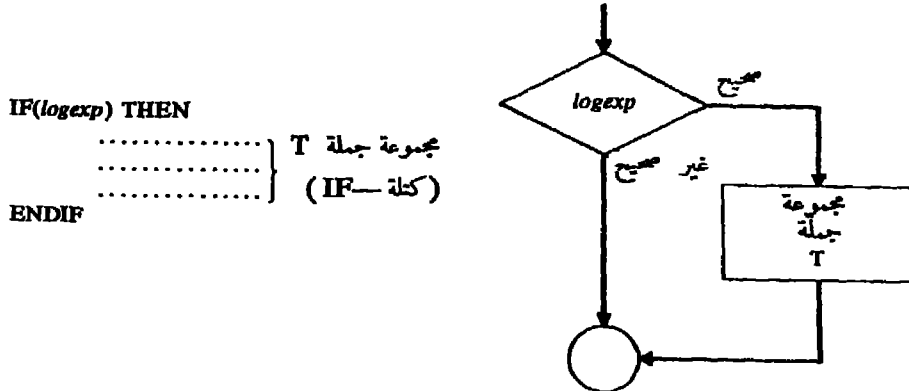
(حيث *logexp* هو تعبير منطقي) وينتهي بجملة

ENDIF

تتنسج هاتان الجملتان كحدود لجسم الهيكل . وسوف نناقش بعضاً من هذه التنوعات .

البديل المفرد

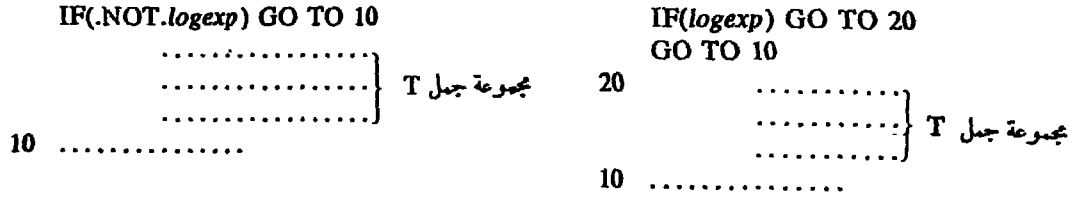
يوضح شكل ١٢ - ١ أحد أشكال I الهيكلية مع خريطة سير العمليات الخاصة به .



شكل ١٢ - ١

أولاً نحسب قيمة التعبير المنطقي *logexp* كما هو موضح في خريطة سير العمليات . إذا كان *logexp* صحيحاً فسوف تنفذ مجموعة جمل T بأكلها (وتسمى كتلة IF—) . أما إذا كان *logexp* غير صحيح ، فسوف تتخطى مجموعة جمل T بأكلها ، وينتقل التحكم إلى أول جملة تلي جملة ENDIF .

يكتاف هيكلي IF— السابق أيأ من أجزاء برامج الفورتران غير الهيكلية التالية والتي تستخدم جملة IF— المنطقية .



نلاحظ أن أول جزء يستخدم مكمل التعبير المنطقي (*logexp*) في جملة IF المنطقية ، وهذا يجب استخدام جملة GO TO إضافية كما في الجزء الثاني . يمكن أن نحصل على مكمل التعبير المنطقي ببساطة بوضع NOT . قبل التعبير . ومع ذلك فكلتات معاملات الترابط الفردية توضح كالاتي :

الأداة	.EQ.	.NE.	.GT.	.GE.	.LT.	.LE.
المكمل	.NE.	.EQ.	.LE.	.LT.	.GE.	.GT.

تذكر أن مكمل المكمل لتعبير منطقي *logexp* هو التعبير المنطقي نفسه ، فمثلاً *logexp* نفسه ، فمثلاً *logexp* . NOT . تكافئ *logexp* .

مسائل ١٢ - ١ :

أجزاء برامج الفورتران التالية متكافئة :

<pre> IF(A.LE.B) GO TO 10 (ب) TEMP = A A = B B = TEMP 10</pre>	<pre> IF(A.GT.B) THEN (أ) TEMP = A A = B B = TEMP ENDIF</pre>
--	---

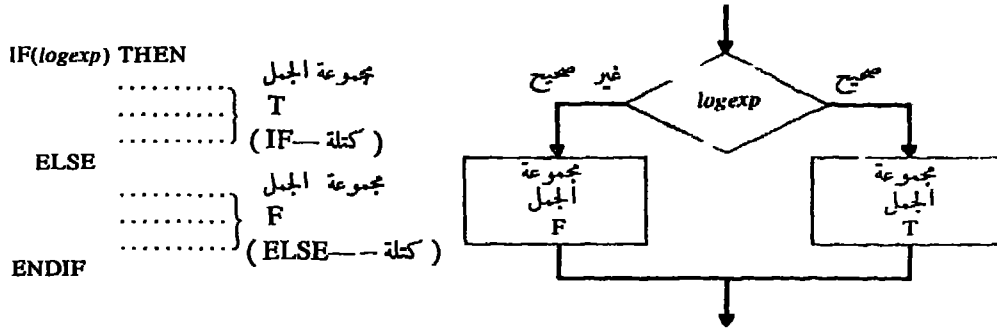
يكتاف هيكلي IF— ذو البديل المفرد والذي يحتوي على جملة واحدة فقط ، جملة IF المنطقية ، والمكس صحيح ، فمثلاً كل ما يأتي متكافئ :

<pre> IF(BIG.LT.A) BIG = A (ب)</pre>	<pre> IF(BIG.LT.A) THEN (أ) BIG = A ENDIF</pre>
--------------------------------------	---

ملاحظة : لايسمح بنقل التحكم إلى داخل كتلة IF— من أي مكان خارج الكتلة نفسها .

البيدليل الثنائي

يمكن أن تصاف جملة ELSE في هيكل IF السابق وذلك لتنفيذ قرارات متعلقة ببديلين مختلفين ، يوضح شكل ١٢-٢ الشكل العام لهذا الهيكل وخريطة سير العمليات الخاصة به .



شكل ١٢-٢

يتم أولاً حساب التعبير المنطقي $logexp$ كما هو موضح بخريطة سير العمليات ، إذا كان التعبير المنطقي $logexp$ صحيحاً تنفذ كتلة IF (مجموعة جمل T) بأكملها . أما إذا كان التعبير المنطقي $logexp$ غير صحيح ، سوف تنفذ مجموعة جمل F (وتسمى كتلة ELSE) بدلا منها . وفي كلتا الحالتين ، بعد الانتهاء من تنفيذ ، إما المجموعة T أو المجموعة F ، سوف ينتقل التحكم إلى أول جملة منفذة بعد جملة النهاية ENDIF .

يكافئ هيكل IF السابق أي من أجزاء برامج الفورتران التالية باستخدام ، جملة IF المنطقية :

<pre>IF(logexp) GO TO 20 GO TO 99 20 99</pre>	(ب)	<pre>IF(.NOT.logexp) GO TO 10 (أ) GO TO 99 10 99</pre>
---	-----	--

لاحظ أنه في حالة عدم استخدام مكل التعبير المنطقي $logexp$ كما في (ب) فإن أماكن مجموعات الجمل سوف تنعكس ، أي ، تظهر مجموعة F أولاً ثم بعد ذلك مجموعة T بينما تظهر مجموعة T أولاً في هيكل IF ثم بعد ذلك مجموعة F .

مثال ١٢-٢

مايل جزءان متكافئان كل يحسب الأجر GROSS لموظف يتقاضى أجراً ونصف إذا عمل ساعات إضافية :

```
IF(HOUR.LE.40.0) THEN (أ)
  GROSS = HOUR*RATE
ELSE
  GROSS = 40.0*RATE + (HOUR - 40.0)*1.5*RATE
ENDIF
```

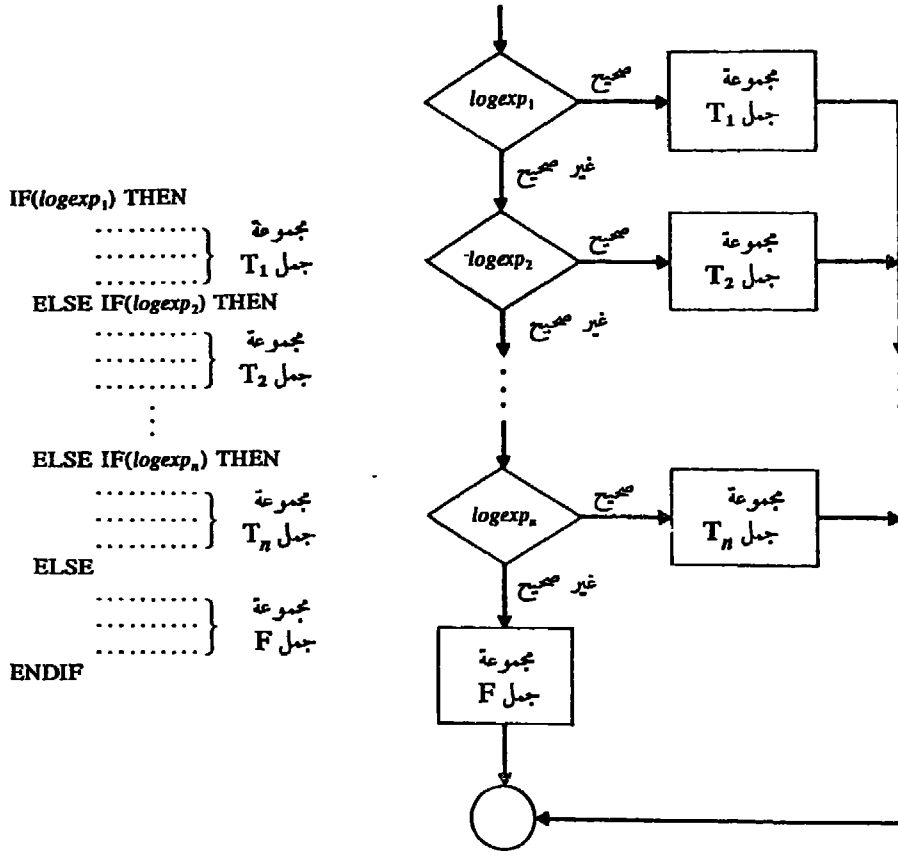
```

IF(HOUR.GT.40.0) GO TO 10
GROSS = HOUR*RATE
GO TO 99
10 GROSS = 40.0*RATE + (HOUR - 40.0)*1.5*RATE
99 .....

```

(ب)

ملاحظة : كما سبق أن ذكرنا فلا يسمح بنقل التحكم إلى داخل كتلة IF أو كتلة ELSE من خارج الكتلة نفسها .



شكل ١٢ - ٢

البدائل المتعددة

يمكننا أيضاً إضافة عدد من جملة ELSE IF بداخل هيكل IF الأساسى لتنفيذ قرارات محددة تشمل أكثر من بدلين . يوضح شكل ١٢ - ٣ مثل هذا الهيكل ذى البدائل المتعددة مع خريطة سير العمليات الخاصة به .

وكما هو موضح في خريطة سير العمليات ، نحسب قيمة التعبيرات المنطقية $logexp_1$ ، $logexp_2$ ، واحداً بعد الآخر . إذا كان أول تعبير منطقي $logexp_k$ صحيحاً ، فسوف تنفيذ مجموعة الجمل T_k . أما إذا كانت كل التعبيرات غير صحيحة ، فسوف تنفذ مجموعة الجمل F . على أى حال ، بعد تنفيذ احدى المجموعات ، سوف ينتقل التحكم إلى أول أمر على جملة ENDIF .

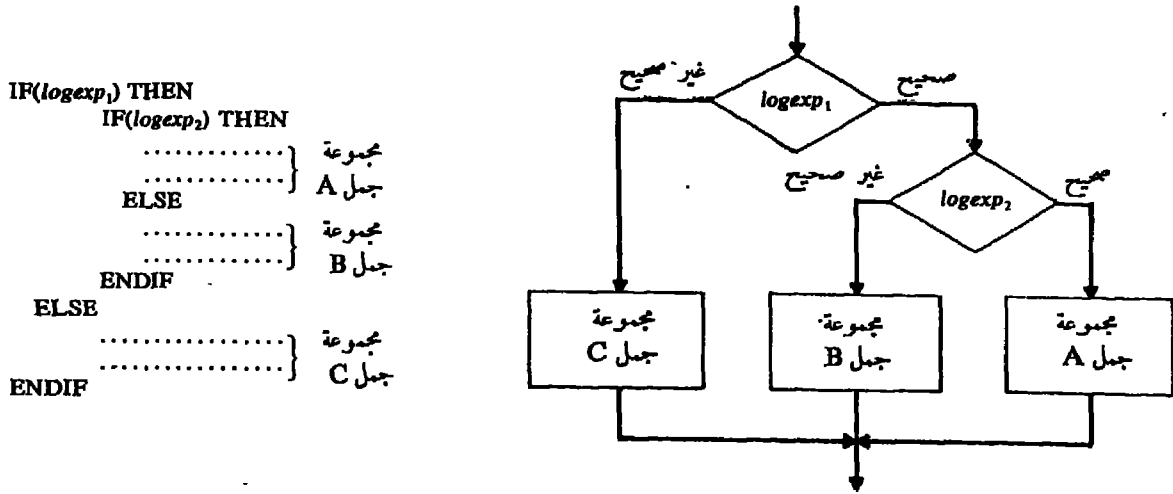
استخدام ELSE اختياري . فثلا تحذف الكلمة الدالة ELSE إن لم تكن هناك جملة F .

مثال ١٢ - ٣

فيما يلي جزء من برنامج الفورتران الهيكل الذي يحسب BONUS لمنوب مبيعات . إذا كتب البرنامج بالفورتران غير الهيكل فسوف تتطلب جمل GO TO كثيرة .

```
IF(SALES.LE.200.0) THEN
  BONUS = 0.0
ELSE IF(SALES.LE.1000.0) THEN
  BONUS = SALES*0.05
ELSE
  BONUS = 50.0 + SALES*0.02
ENDIF
```

أخيراً ، وكما في حلقات DO فلا يمكن لهياكل IF—أن تتداخل . ومع ذلك فيمكن أن يحتوي هيكل IF—على مجموعة كاملة من جمل هيكل IF—أخرى . ويوضح ذلك هيكل البرنامج وخريطة سير العمليات الخاصة به في شكل ١٢ - ٤ .



شكل ١٢ - ٤

١٢ - ٢ هياكل التحكم في الحلقة التكرارية

سنناقش ثلاثة هياكل التحكم في الحلقة التكرارية : حلقة WHILE وحلقة FOR وحلقة DO العامة . تنفذ الحلقتان الأوليتان على كثير من المشغلات ، بما فيها مشغل S—WATFIV والحلقة الثالثة متضمنة في بعض نسخ من فورتران 77 .

حلقة DO الخاصة بالفورتران غير الهيكل والتي نوقشت في الفصل الخامس هيكل مفيد ومناسب للتحكم في الحلقة التكرارية . ومع ذلك فهناك عدة قيود غير مناسبة ، فثلاً يجب أن يكون المتغير الذي يتحكم في الحلقة التكرارية (أى الدليل) متغير من النوع الصحيح و .-ك لا يمكن أن تكون معاملات الحلقة التكرارية تعبيرات حسابية ، وليس في استطاعتنا أيضاً أن نتقص من قيمة الدليل . هذه العوائق أدت إلى إنشاء عدة هياكل لحلقات تكرارية أخرى . ومن الهياكل الأكثر شيوعاً هيكل —WHILE وهيكل —FOR التي سوف نتناقش فيما بعد

هيكل — WHILE

لا يمكن أن نتحكم بسهولة في كثير من العمليات التكرارية بواسطة عداد . ففي مثل هذه الحالات يمكن استخدام هيكل —WHILE بطريقة ملائمة . فيما يلي الشكل العام لهذا الهيكل .

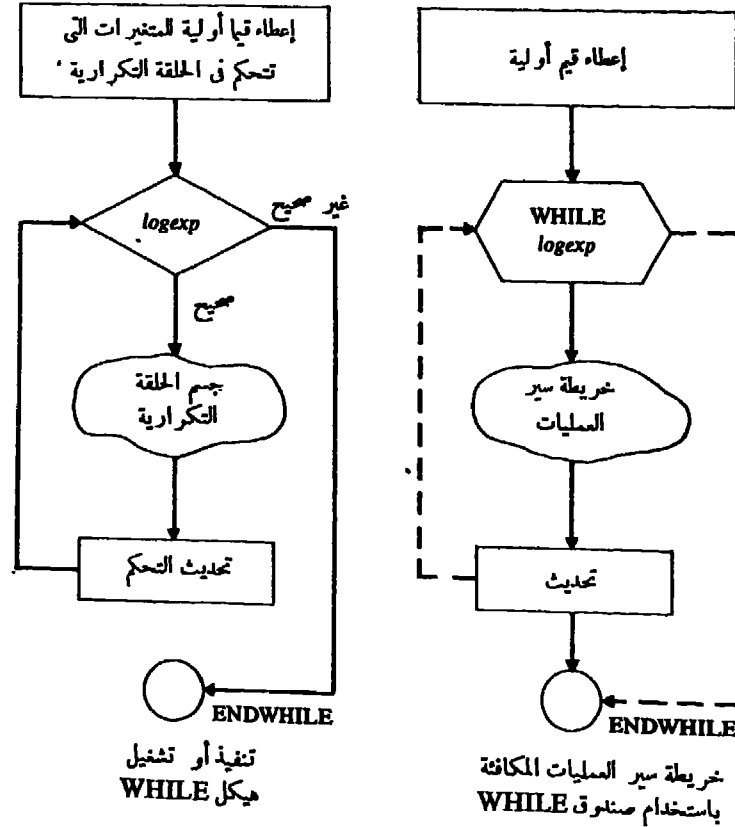
```

WHILE(logexp) DO
    .....
    .....
    .....
ENDWHILE
    
```

جسم الحلقة التكرارية

تُحسب أولاً قيمة التعبير المنطقي $logexp$ ، فإذا كان صحيحاً ، فسوف ينفذ جسم الحلقة التكرارية . وعلاوة على ذلك سيتكرر جسم الحلقة التكرارية ما دام التعبير المنطقي $logexp$ صحيحاً - وسوف تتخطى جسم الحلقة التكرارية عندما يكون التعبير المنطقي $logexp$ غير صحيح وينتقل التحكم إلى أول جملة بعد جملة ENDWHILE . لاحظ أن جملتي WHILE و ENDWHILE تستخدمان كمتحدتين متطابقتين بالهيكل التكراري . وأنه لغاية في الأهمية أن نذكر أننا دائماً نحسب قيمة التعبير المنطقي أولاً قبل تنفيذ الحلقة التكرارية .

عادة يمكن أن يشتمل التعبير المنطقي $logexp$ في هيكل —WHILE على متغير واحد أو أكثر . يجب أن تعطي هذه المتغيرات ، (وتسمى المتغيرات التي تتحكم في الحلقة التكرارية) ، قياً أولية قبل البدء في تنفيذ هيكل —WHILE . ولتجنب وجود حلقة تكرارية لا نهائية ، يجب أيضاً أن تعدل ، أو تحدث قيم المتغيرات مع كل تكرار للحلقة التكرارية ، ويتم التحديث أو التعديل عادة قبل جملة ENDWHILE . ولقد تم توضيح ذلك في شكل ١٢ - ٥ كخريطة سير العمليات التي تمثل هيكل WHILE .



شكل ١٢ - ٥

مثال ١٢ - ٤

أدرس كثيرة الحدود من الدرجة الثانية $2x^2 - 3x - 5$ لا يحسب برنامج المورتان التالى الذى يستخدم هيكل **WHILE** - قيمة x لقيم x التى تراوح ما بين 4 - إلى 4 مخطوات 0.5 .

```

X = -4.0
WHILE(X.LE.4.0) DO
  Y = 2.0*X**2 - 3.0*X - 5.0
  WRITE(6, 10) X, Y
10  FORMAT(2(3X, F10.3))
  X = X + 0.5
ENDWHILE

```

وقارن ببساطة البرنامج السابق بالبرنامج فى المسألة ٥ - ١١ (أ) ، الذى يقوم بعمل نفس الشيء ولكن باستخدام حلقة **DO** الأساسية ذات الدليل . كان يجب فى الحالة السابقة أن تمد مرات التكرارات وأيضاً تكتب X بدلالة الدليل I .

هيكل **FOR** -

وكما ذكرنا سابقاً فإن أحد القيود الصارمة لحلقة **DO** غير الهيكلية هو عدم مقدرتها على إنقاص قيمة الدليل . عائق آخر وهو عدم اختيار الدليل لقيمة النهاية إلا بعد أول تكرار . وعلى ذلك فرغم أن $M > N$ فإن حلقة **DO** الأساسية (غير الهيكلية) المبتدئة بجملة **DO** التالية :

```
DO 50 K = M, N
```

سننفذ مرة واحدة .

لقد تخلصنا من هذه العوائق فى هيكل **FOR** - والتى لها الشكل التالى :

```
FOR(INDEX = INV, ENDV, INCR) DO
```

```

.....
.....
.....
.....
.....

```

(١) جسم الحلقة التكرارية

```
ENDFOR
```

الدليل **INDEX** يجب أن يكون إسماً متغيراً صحيحاً . ومع ذلك فإن كلا من **INV** و **ENDV** و **INCR** والتى تمثل القيمة الابتدائية وقيمة النهاية ومعامل الزيادة على الترتيب ، يمكن أن تكون أى تعبير صحيح فيما عدا أن معامل الزيادة **INCR** لا يمكن أن يأخذ القيمة صفراً .

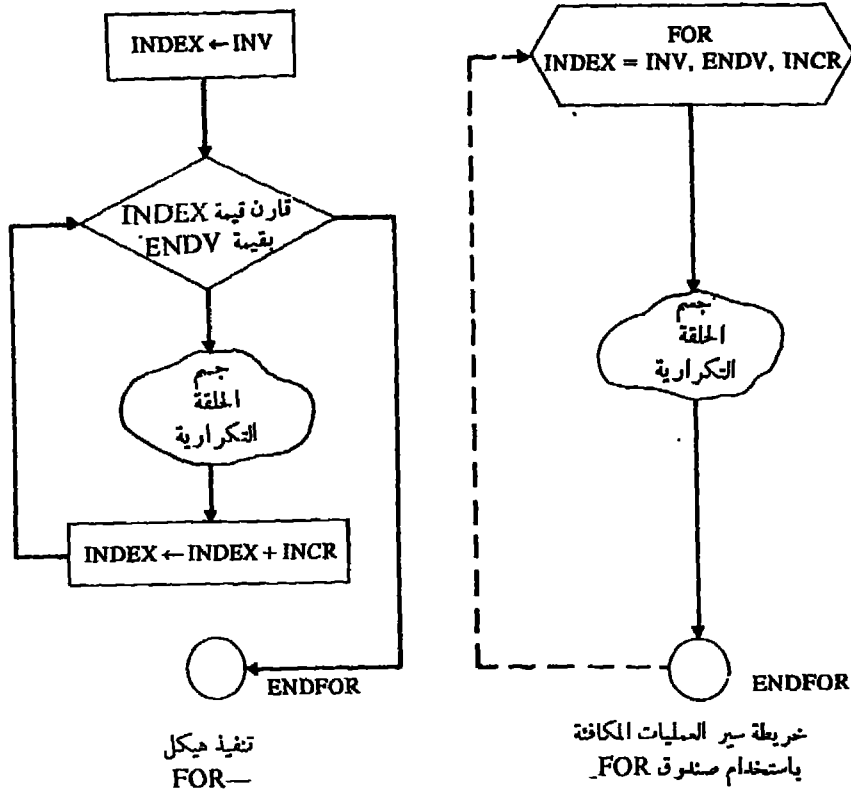
يوضح شكل ١٢ - ٦ خريطة سير العمليات لتنفيذ هيكل **FOR** - ، نجمل قيمة الدليل **INDEX** فى البداية مساو **INV** . نختبر قيمة **INDEX** قبل تنفيذ الحلقة التكرارية ، لئرى ما إذا تمدت قيمة النهاية المطاة **ENDV** . يختلف هذا الاختبار حسب قيمة **INCR** هل موجبة أم سالبة ؟ إذا كانت **INCR** موجبة ، يكون السؤال :

is $INDEX \leq ENDV$?

ومع ذلك فإذا كانت **INCR** سالبة ، يكون السؤال :

is $INDEX \geq ENDV$?

نستمر فى تنفيذ الحلقة التكرارية طالما لم تمتد قيمة **INDEX** القيمة **ENDV** أى طالما تكون نتيجة الاختبار « نعم » .



شكل ١٢ - ٦

تزداد قيمة INDEX بمقدار INCR بعد تنفيذ الحلقة التكرارية وقبل التكرار التالى .

حيث أن معامل الزيادة INCR يمكن أن يكون سالباً ، لذلك (يمكن إنقاص قيمة INDEX) والمثال التالى يوضح هذه الميزة :

مثال ١٢ - ٥

افرض أننا نريد طباعة الأعداد الصحيحة 100 و 99 و 98 و ... و 1 وبهذا الترتيب يمكننا عمل ذلك بطريقتين : (١) بحلقة — DO الأساسية (غير الميكالية) (أنظر مثال ٥ - ٥) و (٢) بهيكل — FOR .

```

DO 500 K = 1, 100      (ب)      FOR(K = 100, 1, -1) DO (أ)
  J = 101 - K          WRITE(6, 30) K
  WRITE(6, 30) J       30      FORMAT(1X, I5)
30      FORMAT(1X, 5I)  ENDFOR
500 CONTINUE

```

لاحظ أن الحلقة التكرارية في (ب) تنتهى عندما تأخذ K القيمة 0 أى بعد أن تتجاوز قيمة النهاية 1 .

حلقة DO المعممة (حلقة — DO الميكالية)

تتجاهل هيكل — WHILE وهيكل — FOR بعض العوائق الموجودة في حلقة — DO الأساسية (غير الميكالية) التي تمت مناقشتها في الفصل الخامس . يحتوى فورتران 77 الجديد بناء حلقة — DO المعممة التي تجمع خصائص كل من هيكل — WHILE وهيكل — FOR في بناء واحد .

وحلقة DO- المممة هذه نفس المظهر الأساسي لحلقة DO- ذات الدليل التي تم عرضها في الأقسام ٥-٣ و ٥-٤ وسيكون لها الشكل التالي :

```
DO n VAR = INV, ENDV, INCR
.....
.....
.....
n CONTINUE
```

} جسم حلقة - DO

مرة ثانية نحيط جسم الحلقة التكرارية المراد تكراره بزواج الجمل DO-CONTINUE .

يسمى VAR متغير التحكم في الحلقة ، ويمكن أن يكون اسم متغير صحيح أو اسم متغير حقيقي ، تمثل معاملات الحلقة التكرارية INV و ENDV و INCR القيمة الابتدائية والقيمة النهائية (أو قيمة الاختيار) ومعامل الزيادة على الترتيب ، ويمكن أن تكون أي تميرات صحيحة أو حقيقية . (وهذا يتعارض مع حلقة - DO المشروحة في الفصل الخامس حيث يجب أن يكون الدليل متغيراً صحيحاً وأن تكون المعاملات ثوابت صحيحة موجبة أو متغيرات صحيحة) . وإن لم يكن INCR موجوداً ، فتعتبر معامل الزيادة 1 ، ولذلك ، لا يمكن أن تكون INCR له قيمة صفراً .

من المناقشة في الفصل الخامس وتبعاً لهيكل FOR- السابق فإن الصيغة DO=CONTINUE السابقة تقترح أن نبدأ التكرار بجعل قيمة VAR الابتدائية مساوية لقيمة INV ، ثم تزداد قيمة VAR بمقدار INCR مع كل تكرار ، وتستمر في تكرار الحلقة حتى تتجاوز قيمة VAR قيمة ENDV . هذا التفسير صحيحاً أساساً ، ومع ذلك فن الضروري إعطاء معنى دقيق لهذا البناء الهام أي حلقة DO- المممة .

تظهر خريطة سير العمليات لتنفيذ حلقة DO- المممة في شكل ١٢-٧ ، عند تنفيذ جملة DO- بحسب قيم INV و ENDV و INCR ثم تخصص القيمة الابتدائية INV للمتغير VAR وتخصص قيمة عداد التكرار والذي رمزنا له بالرمز KOUNT . وعادة يمكن أن نحسب عداد التكرار KOUNT كنتيجة للقيمة الصحيحة $(ENDV - INV + INCR)/INCR$ (تقريباً) :

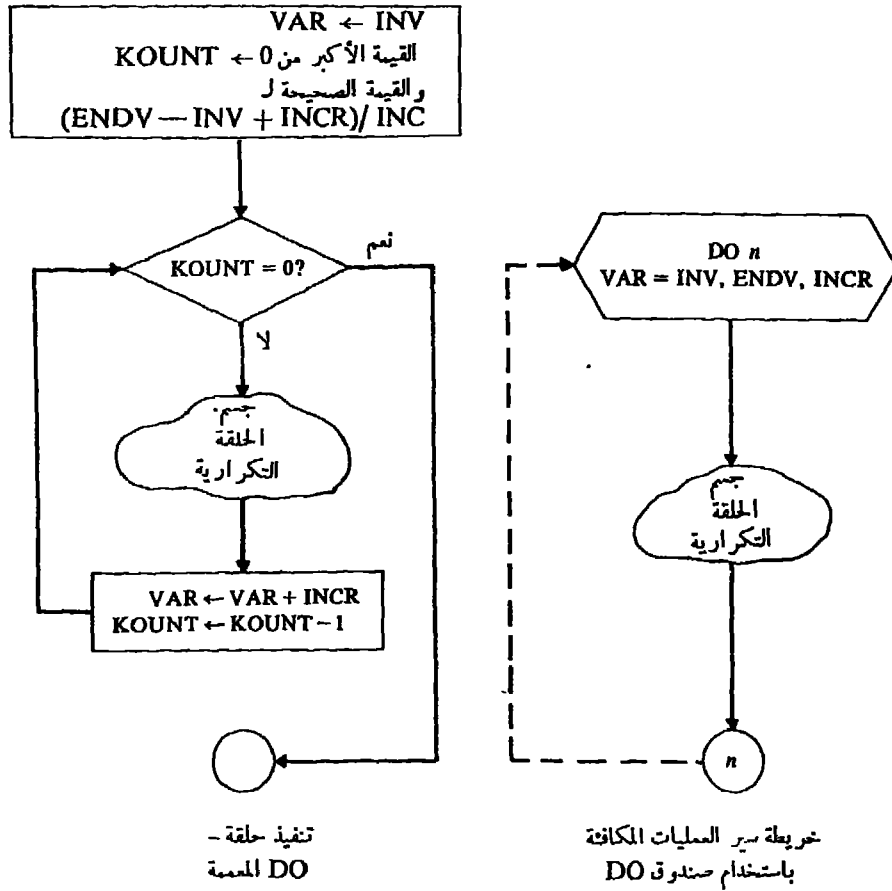
KOUNT = 4 تعطى DO 100 K = 2, 13, 3
KOUNT = 21 تعطى DO 200 K = 2.0, 1.0, -0.05

لعمل حساب المواضع التي يكون فيها $INV > ENDV$ و INCR موجبة ، أو عندما تكون $INV < ENDV$ و INCR سالبة (أي ، عندما تتجاوز قيمة VAR من البداية قيمة النهاية ENDV) ، تخصص إلى KOUNT القيمة الأكبر من الصفر والقيمة الصحيحة لـ $(ENDV - INV + INCR)/INCR$. ونؤكد أن هذا العدد KOUNT يعطى عند المرات التي ستنفذ فيها الحلقة التكرارية .

وكما هو موضح بالشكل ١٢ - ٧ نختبر ما إذا كانت قيمة KOUNT تساوي صفر قبل تنفيذ الحلقة التكرارية . إذا كانت KOUNT صفر فستنفذ الحلقة التكرارية وتزداد قيمة VAR بمقدار INCR وتنقص عداد التكرار KOUNT بمقدار واحد . وينتقل التحكم إلى أول جملة منفذة تلي جملة CONTINUE عندما تصبح قيمة KOUNT = 0 .

مثال ١٢ - ٦

(أ) افترض أننا نريد أن نطبع الأعداد الصحيحة 100 و 99 و ... و 1 بهذا الترتيب . فباستخدام حلقة DO- المممة يكون لدينا مايل :



شكل ١٢-٧

```

DO 500 K = 100, 1, -1
    WRITE(6, 30).K
30    FORMAT(1X, I5)
500  CONTINUE

```

(ب) وباستخدام حلقة DO المعممة لحل المسألة في مثال ١٢ - ٤ يكون لدينا مايل :

```

DO 999 X = -4.0, 4.0, 0.5
    Y = 2.0*X**2 - 3.0*X - 5.0
    WRITE(6, 10) X, Y
10    FORMAT(2(3X, F10.3))
999  CONTINUE

```

يمكن تلخيص خصائص DO المعممة كالآتي :

- ١ - كما هو متعارف إذا لم توجد INCR فتمتد مقدار الزيادة 1 .
- ٢ - ليس من الضروري أن تكون أنواع قيم معاملات الحلقة التكرارية ENDV و INCR و INV من نفس نوع المتغير VAR . في مثل هذه الحالة تحول أولاً قيم المعاملات إلى نوع المتغير VAR .
- ٣ - يمكن أن تكون قيمة INCR سالبة وعلى ذلك من الممكن أن تنقص قيمة VAR . على أى حال، فإننا نستمر في تنفيذ الحلقة التكرارية لكل قيم VAR إلى أن تتجاوز قيمة VAR قيمة النهاية ENDV .

- ٤ - لا يمكن تغيير قيمة متغير الحلقة التكرارية VAR بداخل جسم حلقة DO إلا بطريقة الزيادة الخاصة به .
- ٥ - يعرف دائماً متغير الحلقة التكرارية عندما ينتقل التحكم من حلقة DO- ، (يتعارض هذا مع دليل حلقة DO الأساسية (غير الهيكلية) ، وفيها يعرف فقط في حالة الخروج غير الطبيعي من الحلقة التكرارية) .
- ٦ - إذا كانت قيمة $INV > INCR$ موجبة ، أو إذا كانت $INV < ENOV$ سالبة سيكون للمداد KOUNT قيمة ابتدائية صفر ولن ينفذ جسم الحلقة التكرارية على الإطلاق .
- ٧ - لا يسمح بنقل التحكم إلى وسط حلقة DO المممة من الخارج (كما في حلقة DO- الأساسية ذات الدليل) . تطبق هنا أيضاً قواعد حلقات DO المتداخلة التي تم مناقشتها في قسم ٥ - ٩ .
- ٨ - لا يمكن أن تتداخل حلقات DO وهايكل IF- ، ومع ذلك ، فيمكن أن تحتوي كتلة جملة هيكل IF على حلقة DO . ويمكن أن تحتوي حلقة DO على هيكل IF- .

ملاحظة : إذا كان متغير الحلقة التكرارية VAR متغيراً صحيحاً ، فإن تنفيذ حلقة DO التكرارية المممة يشبه هيكل FOR- التي تمت مناقشته سابقاً . وفي الحقيقة تتصرف حلقة DO المممة بطريقة ماثلة جداً لحلقة DO الأساسية ذات الدليل فيما عدا : (أ) أنه يمكن أن تكون INCR سالبة ، (ب) ويمكن أن تكون معاملات الحلقة التكرارية تعبيرات (ج) وأن الاختيار يتم قبل تنفيذ الحلقة التكرارية .

مسائل محلولة

هايكل IF-

١٢ - ١ افترض أن نسبة RATE الفائدة على قرض AMT هي 7% إذا كان القرض $AMT \leq \$10,000,00$. وتكون النسبة 6% إذا تعدت AMT مبلغ $\$10,000.00$ بفرض أن INT متغير حقيقي وأن AMT مخزنة في الذاكرة ، اكتب البرنامج الذي يحدد قيمة INT على القرض .

يحتوي شكل ١٢ - ٨ مثل هذا البرنامج (قارنه مع البرنامج غير الهيكل على صفحة ١٠١)

```

IF(TYPE.EQ.1) THEN
  NET = PAY - 9.75
ELSE IF(TYPE.EQ.2) THEN
  NET = PAY - 16.25
ELSE IF(TYPE.EQ.3) THEN
  NET = PAY - 24.50
ENDIF
WRITE(6, 20) ID, NET
20 FORMAT(1X, I5, 3X, F12.2)

IF(AMT.LE.10000.0) THEN
  RATE = 0.07
ELSE
  RATE = 0.06
ENDIF
INT = AMT*RATE
WRITE(6, 10) AMT, RA  INT
10 FORMAT(3(5X, F10.2))

```

شكل ١٢ - ٩

شكل ١٢ - ٨

١١ - ٢ افترض أن قسط التأمين الصحي يخصم من مرتب موظف تبعاً للقطعة التالية (حيث TYPE متغير صحيح) :

9.75 إذا كانت 1 TYPE = (أعزب)
 16.25 إذا كانت 2 TYPE = (متزوج ولا يعمل)
 24.50 إذا كانت 3 TYPE = (متزوج ويعمل)

القسط =

اكتب جزء برنامج يخصم القسط من مرتب موظف بفرض أن PAY و TYPE و ID مخزنة في الذاكرة (وأن NET متغير حقيقي) .

تظهر خريطة سير العمليات لمثل هذا البرنامج وترجمته إلى الفورتران غير الهيكل على صفحة ١٠٨ . يحتوى شكل ١٢ - ٩ ترجمة الفورتران الهيكل باستخدام هيكل IF-ذى البدائل المتعددة .

١٢-٣ اكتب جزء البرنامج الذى يقرأ عدد صحيحاً M ويطبع ما إذا كان هذا العدد موجباً أو سالباً أو صفر .
يحتوى شكل ١٢ - ١٠ على هذا البرنامج باستخدام هيكل IF-ذى البدائل المتعددة .

```

READ(5, 10) A, B, C
10 FORMAT(3F10.2)
WRITE(6, 20) A, B, C
20 FORMAT(1X, 'SIDES', 3(2X, F10.2))
IF(A.EQ.B.AND.B.EQ.C) THEN
  WRITE(6, 30)
30   FORMAT(1X, 'EQUILATERAL')
ELSE IF(A.EQ.B.OR.A.EQ.C.OR.B.EQ.C) THEN
  WRITE(6, 40)
40   FORMAT(1X, 'ISOSCELES')
ENDIF

```

شكل ١٢ - ١١

```

READ(5, 10) M
10 FORMAT(15)
WRITE(6, 20) M
20 FORMAT(1X, I5, 2X, 'IS')
IF(M.GT.0) THEN
  WRITE(6, 30)
30   FORMAT(1X, 'POSITIVE')
ELSE IF(M.EQ.0) THEN
  WRITE(6, 40)
40   FORMAT(1X, 'ZERO')
ELSE
  WRITE(6, 50)
50   FORMAT(1X, 'NEGATIVE')
ENDIF

```

شكل ١٢ - ١٠

١٢-٤ اكتب جزء البرنامج الذى يقرأ الأطوال A و B و C لمثلث T ويطبع ما إذا كان T مثلث متساوى الأضلاع أو مثلث متساوى الساقين .

يكون T مثلث متساوى الأضلاع إذا كانت $A = B$ و $A = C$ و $B = C$. ويكون T مثلث متساوى الساقين إذا كانت $A = B$ أو $A = C$ أو $B = C$. يحتوى شكل ١٢ - ١١ على مثل هذا البرنامج (نستخدم الروابط المنطقية AND. و OR. التى تم مناقشتها فى الفصل التاسع) .

هياكل-DO.

١٢-٥ أوجد عداد التكرار KOUNT لكل جملة DO التالية :

(أ) DO 100 K = 4, 20, 3
(ب) DO 200 X = -4, 4, 0.6
(ج) DO 300 K = 4, -9, -2
(د) DO 400 X = 5, -2, 0.2

أولا أوجد خارج القسمة $Q = (ENDV - INV + INCR)/INCR$. إجمال J ترمز إلى القيمة الصحيحة Q . تكون بعد ذلك KOUNT مساوية J أو 0 أيهما أكبر .

$$(أ) Q = (20 - 4 + 3)/3 = 19/3 ; \text{ إذن } J = 6 \text{ من ثم } KOUNT = 6$$

$$(ب) Q = (4 + 4 + 0.6)/0.6 = 8.6/0.6 ; \text{ إذن } J = 14 \text{ من ثم } KOUNT = 14$$

$$(ج) Q = (-9 - 4 - 2)/2 = -15/-2 ; \text{ إذن } J = 7 \text{ من ثم } KOUNT = 7$$

$$(د) Q = (-2 - 5 + 0.2)/0.2 = -6.8/0.2 ; \text{ إذن } J = -34 \text{ من ثم } KOUNT = 0$$

١٢-٦ أوجد القيمة النهائية لكل من K و L .

```
DO 200 K = 5, 1 (ب)
    L = K**2
200 CONTINUE
    K = 2*K
```

```
DO 100 K = 1, 5 (أ)
    L = K**2
100 CONTINUE
    K = 2*K
```

(أ) تنفذ حلقة DO لقيم $K = 1, 2, 3, 4$ ثم للقيمة 5 من ثم تكون القيمة النهائية لـ L هي $5^2 = 25$. ومع ذلك فعندما ينتقل التحكم من الحلقة التكرارية ، فتراد قيمة K وتصيح 6 . من ثم تكون قيمة K النهائية 12 .

(ب) لا تنفذ حلقة DO على الإطلاق حيث أن عدد التكرار $KOUNT = 0$ ؛ من ثم تكون L غير معرفة . ومع ذلك ، تخصص القيمة الابتدائية لـ K عند تنفيذ جملة DO . من ثم تكون قيمة K النهائية 10 .

١٢-٧ افرض أن $z = x^2 - y^3$ اكتب جزء البرنامج الذى يوجد z لقيم x و y بحيث تتغير كل من x و y من 4 - إلى 4 بخطوات مقدارها 0.5 .

يحتوى شكل ١٢-١٢ على مثل هذا البرنامج الذى يستخدم حلقات DO المعممة المتداخلة (قارن هذا البرنامج بالبرنامج المشابه فى المسألة ١١-٥ (ب)) .

```
DO 200 X = -4, 4, 0.5
    DO 100 Y = -4, 4, 0.5
        Z = X**3 - Y**2
        WRITE(6, 10) X, Y, Z
        FORMAT(3(2X, F10.2))
    100 CONTINUE
200 CONTINUE
```

```
FOR(J = N, K, -1) DO
    A(J + 1) = A(J)
ENDFOR
A(K) = D
```

شكل ١٢-١٣

شكل ١٢-١٢

١٢-٨ اكتب جزء البرنامج باستخدام هيكل FOR- الذى يضيف عنصر D فى المكان K من مجموعة متراسة $A(1), \dots, A(2), \dots, A(N)$

كما تمت مناقشته فى مسألة ٦-١٠ ، يجب أن نحرك جزء المجموعة المتراسة $A(K)$ و \dots و $A(N)$ إلى أسفل مكان واحد قبل تخصيص D إلى $A(K)$. يتم هذا بتخصيص $A(N)$ إلى $A(N + 1)$ أولاً ، ثم $A(N - 1)$ إلى $A(N)$ وهكذا إلى أن نصل $A(K)$ إلى $A(K + 1)$. يحتوى شكل ١٢-١٣ على مثل هذا البرنامج .

١٢-٩ اكتب برنامجاً باستخدام هيكل WHILE- الذى يطبع كل الأعداد الصحيحة الموجبة المفردة أقل من 100 مع حذف مضاعفات 7 : 1, 3, 5, 9, 11, 13, 15, 17, 19, 23, \dots, 97, 99

يحتوى شكل ١٢-١٤ على مثل هذا البرنامج . لاحظ كيف تعطى N قيمة ابتدائية قبل الحلقة التكرارية وكيف يتم تعديلها (تعديلها) عند نهاية الحلقة التكرارية .

```
DIMENSION SCORE(300)
N = 1
READ(5, 10) GRADE
10 FORMAT(F8.0)
WHILE(GRADE.GE.0.0) DO
    SCORE(N) = GRADE
    N = N + 1
    READ(5, 10) GRADE
ENDWHILE
N = N - 1
```

```
N = 1
WHILE(N.LT.100) DO
    K = (N*7)/7
    IF(K.EQ.N) GO TO 5
    FORMAT(6, 10)
    10 FORMAT(10X, I2)
    5 N = N + 2
ENDWHILE
```

شكل ١٢-١٥

شكل ١٢-١٤

١٢ - ١٠ تم تقييم درجات اختبار لطلبة درجة واحدة على كل بطاقة ، والمجموعة لها بطاقة خلفية مثقبة عليها رقم سالب . بفرض وجود أقل من 300 درجة ، أكتب جزء برنامج باستخدام هيكل WHILE الذي يحسب عدد الدرجات N ويخزنها في مجموعة متراسة SCORE .

يحتوي شكل ١٢ - ١٥ على مثل هذا البرنامج . لاحظ أن N تنقص بمقدار واحد سد نقل التحكم من الحلقة التكرارية (لماذا ؟) .

مسائل تكميلية

هياكل IF- و DO-

١٢ - ١١ افرض أن $J = 5$ و $K = 10$. أوجد القيمة النهائية لكل من J و K إذا نفذنا ما يلي :

(أ)
IF(J.GT.10) THEN
 $K = K + 5$
 $J = J + K$

ENDIF
 $J = J + K$
 $J = 2 * J$

(ب)
IF(J.GT.10) THEN
 $K = K + 5$
 $J = J + K$
ELSE
 $J = J + K$
ENDIF
 $J = 2 * J$

(أ)
IF(J.LE.10) THEN
 $K = K + 5$
 $J = J + K$

ENDIF
 $J = J + K$
 $J = 2 * J$

(ب)
IF(J.LE.10) THEN
 $K = K + 5$
 $J = J + K$
ELSE
 $J = J + K$
ENDIF
 $J = 2 * J$

١٢ - ١٢ أدرس أجزاء البرامج التالية :

(أ)
 $M = 5$
IF(J.LT.K) THEN
 $J = J + 5$
 $M = M + 3$
ELSE IF(J.LT.8) THEN
 $M = M + 10$
ELSE
 $M = M + 20$
ENDIF

(ب)
 $M = 5$
IF(J.LT.K) THEN
 $J = J + 5$
 $M = M + 3$
IF(J.LT.8) THEN
 $M = M + 10$
ELSE
 $M = M + 20$
ENDIF

أوجد قيمة M النهائية لكل برنامج إذا كانت :

(أ) $J = 2, K = 5$ (ب) $J = 2, K = 1$ (ج) $J = 10, K = 15$ (د) $J = 10, K = 1$

١٢ - ١٣ أوجد عدد التكرار KOUNT لكل جملة من جمل DO التالية :-

(أ) DO 300 K = 2, 25, -6
(ب) DO 400 X = 1, -3, -0.2

(أ) DO 100 K = -3, 19, 4
(ب) DO 200 X = 4, -6, -0.3

١٢-١٤ أوجد القيمة النهائية لكل من J و M إذا نفذنا ما يلي :

<p>DO 300 J = 9, 2, 3 (ـ) M = J + 5 300 CONTINUE DO 400 J = 2, 3, 9 (د) M = J + 5 400 CONTINUE</p>	<p>DO 100 J = 2, 9, 3 (أ) M = J + 5 100 CONTINUE DO 200 J = 9, 2, -5 (ب) M = J + 5 200 CONTINUE</p>
--	---

١٢-١٥ أوجد القيمة النهائية لكل من J و K و L بعد تنفيذ ما يلي :

<p>J = 3 (ب) WHILE(J.LT.6) DO K = 2*J L = 3*J J = J + 2 ENDWHILE</p>	<p>DO 100 J = 2, 6, 3 (أ) DO 200 K = J, 4 L = J + K 200 CONTINUE 100 CONTINUE</p>
--	---

برامج :

١٢-١٦ يخصص لدرجة الاختبار T الحرف A أو B أو C أو D أو F تبعاً لما يأتي :

$T \geq 90$ أو $80 \leq T < 90$ أو $70 \leq T < 80$ أو $60 < T < 70$ أو $T < 60$

اكتب البرنامج الذي يقرأ درجة الاختبار ويطبع الحرف المقابل له .

١٢-١٧ تحتوي كل بطاقة في المجموعة درجة اختبار الطالب ، والمجموعة لها بطاقة (خلفية) مثقب بها رقم سالب . اكتب البرنامج الذي يجد عدد الطلبة N الذين أدوا الاختبار وعدد الطلبة M الذين كانت أوراقتهم ممتازة (أى ، أخذوا 100) ، وعدد الطلبة L الذين رسبوا (أى حصلوا على $60 >$) .

١٢-١٨ اكتب البرنامج الذي يحسب الجذور الحقيقية لمعادلة الدرجة الثانية :

$$ax^2 + bx + c = 0$$

مع معلومية المعاملات A و B و C حيث $0 \neq A$ (قارن مع البرنامج على صفحة ١٠٩) .

١٩-١ اكتب البرنامج الذي يقرأ ثلاثة أرقام موجبة A و B و C ويحدد ما إذا كانت A و B و C أطوال أضلاع مثلث . وإذا كانت الإجابة نعم احسب محيط المثلث ، أما إذا كانت الإجابة لا اطبع الرسالة « NOT A TRIANGLE » (ليس مثلثاً) . (تلميح : لا يمكن أن تشكل A و B و C مثلثاً إلا إذا كان طول ضلع ما أكبر من أو يساوى مجموع الضلعين الآخرين) . اختبر البرنامج بالبيانات :

(أ) $A = 3.0, B = 4.0, C = 5.0$ (ب) $A = 3.0, B = 12.0, C = 5.0$

٢٠-١ اكتب برنامجاً باستخدام حلقة DO معمة يضيف عنصر D في المكان K في مجموعة مناصرة A(1) و A(2) و ... و A(n) (قارن مع المسائل ١٢-٨ و ١٢-١٠)

١٢ - ٢١ أوجد عدد النقط ذات الاحداثيات الصحيحة والتي تقع داخل :

$$(أ) \text{ الدائرة } 50 = x^2 + y^2 \text{ (ب) القطع الناقص } 100 = 3y^2 + 2x^2 \text{ (تارن مع المسألة ٥ - ١٢) .}$$

١٢ - ٢٧ اكتب البرنامج الذي يطبع الرقم 20 عشرين مرة والرقم 19 تسع عشرة مرة والرقم 18 ثمان عشرة مرة ، وهكذا .

١٢ - ٢٣ تحتوي كل بطاقة في مجموعة بطاقات على عدد صحيح موجب . والمجموعة لها بطاقة خلفية خالية . اكتب البرنامج الذي يحدد عدد الأرقام الصحيحة الزوجية وعدد الأرقام الصحيحة الفردية .

١٢ - ٢٤ إدرس المعادلة $y = x^2 - 4x + b$. اكتب البرنامج الذي يحدد لا لقيم x إبتداء من I إلى J بزيادة قدرها D ، وحيث I و J و D مثبتة على بطاقة و $I < J$ و $D > 0$.

١٢ - ٢٥ إفرض أنك أودعت مبلغ \$2,000.00 في حساب توفير يعطى 7% فائدة مركبة سنوياً . اكتب برنامجاً يستخدم هيكل WHILE لتحديد عدد السنوات التي تنقضي ليصل الحساب إلى \$5,000.00 .

١٢ - ٢٦ قررت إدارة محل أن تعطى خصماً على ثمن PRICE للتليفزيون الملون تبعاً لحجم الجهاز SIZE بالبوصة كما يلي :

\$ 40,00	إذا كان الحجم 12 أو 15	} الخصم =
\$ 70.00	إذا كان الحجم 17 أو 19	
\$ 110.00	إذا كان الحجم 21 أو 23	
\$ 140.00	إذا كان الحجم 25	

هناك خصم إضافي مقداره 3% من الثمن PRICE إذا كان $400 < \text{PRICE} \leq 300$ و 4% إذا كان $\text{PRICE} > 400$. اكتب البرنامج الذي يقرأ الرقم ID لكل TV وكذا حجمه SIZE و ثمنه PRICE ويحدد ثمن TV الصافي NET (إفرض أن SIZE متغير صحيح و NET متغير حقيقي)

١٢ - ٢٧ هناك أربعة مرشحين لمنصب في مجلس مدرسة . يدلي الشخص بصوته بواسطة تثقيب I أو 2 أو 3 أو 4 على بطاقة . سيكون لمجموعة البطاقات بطاقة خلفية مثبت عليها 1 . اكتب البرنامج الذي يحصى عدد الأشخاص N الذي أدلوا بأصواتهم ، وعدد الأصوات - التي حصل عليها كل مرشح .

١٢ - ٢٨ أعد كتابة المسألة ١٢ - ٢٧ ، غير أن البرنامج هنا يحدد أيضاً الفائز في الانتخابات .

١٢ - ٢٩ أعد كتابة ١٢ - ٢٧ غير أن في هذه المرة عدد المرشحين لمنصبين خمسة والبرنامج يحدد أيضاً الفائزين .

حلول المسائل المختارة

١١-١٢

30, 10 (د) 30, 10 (ـ) 40, 15 (ب) 70, 15 (أ)

١٢-١٢

8 (٢) 28 (١) (ـ) 8 (ii) 18 (i) (أ)

25 (٢) 25 (١) (د) 15 (ii) 15 (i) (ب)

١٣-١٢

20 (د) 0 (ـ) 33 (ب) 5 (أ)

J = 9, M (ليست معرفة) (ـ)

J = 11, M = 13 (أ) ١٤-١٢

J = 11, M = 7 (د)

J = -1, M = 9 (ب)

١٥-١٢

J = 7, K = 10, L = 15 (ب)

J = 8, K = 5, L = 4 (أ)

ملحق (١)

دوال المكتبة

يمطى الملحق قائمة لدوال المكتبة (الميئة - أو المبنية داخلياً) والمتوافرة لمعظم نظم الفورتران . يشار إلى خلاصات الدالة في جدول أ - ١ بالرموز التالية . والتي تدل على أنواعها .

K و J نوع صحيح
X و Y نوع حقيقي
DX و DY نوع متضاعف الدقة
C نوع مركب

سنستخدم كلمة متضاعفة (double) للدقة المتضاعفة . يرمز إلى إشارة الخلاصة a بالرمز $\text{sgn}(a)$ وتعرف بالآتي :

تكون $\text{sgn}(a) = 1$ إذا كانت $a \geq 0$

تكون $\text{sgn}(a) = -1$ إذا كانت $a < 0$

يمز إلى دالة أكبر عدد صحيح بالرمز $[a]$ وهي تمطى أكبر عدد صحيح أقل أو يساوى a

جدول أ - ١

نوع الدالة	اسم الدالة بالخلاصات	التعريف	الوصف الكلاسي
حقيقي دقة متضاعفة مركب	SQRT(X) DSQRT(DX) CSQRT(C)	\sqrt{a}	جذر تربيعي
حقيقي دقة متضاعفة مركب	EXP(X) DEXP(DX) CEXP(C)	e^a	أس
حقيقي دقة متضاعفة مركب	ALOG(X) DLOG(DX) CLOG(C)	$\log_e(a)$	لوغاريتم طبيعي (للأساس e)
حقيقي دقة متضاعفة	ALOG10(X) DLOG10(DX)	$\log_{10}(a)$	لوغاريتم عادي (للأساس 10)
صحيح حقيقي دقة متضاعفة	IABS(J) ABS(X) DABS(DX)	$ a $	قيسة مطلقة
حقيقي مركب حقيقي حقيقي	CABS(C) CONJG(C) REAL(C) AIMAG(C)	$\sqrt{x^2 + y^2}$ $x - iy$ x y	دوال ذات خلاصة مركبة $C = x + iy$ القيمة مواقع مركب جزء حقيقي جزء تخيلى

جدول أ - ١ (تابع)

نوع الدالة	اسم الدالة بالاختصارات	التعريف	الوصف الكلامي
مركب	CMPLX(X, Y)	$x + iy$	الشكل المركب لرقين حقيقيين (x, y)
حقيقي دقة متضاعفة دقة متضاعفة حقيقي مصحح	FLOAT(J) DFLOAT(J) DBLE(X) SNGL(DX) IFIX(X)	تحويل صحيح إلى حقيقي تحويل صحيح إلى دقة متضاعفة تحويل حقيقي إلى دقة متضاعفة تحويل دقة متضاعفة إلى حقيقي بالبت تحويل حقيقي إلى صحيح بالبت	تحويل النوع
مصحح مصحح حقيقي حقيقي دقة متضاعفة	MAX0(J, K, ...) MAX1(X, Y, ...) AMAX0(J, K, ...) AMAX1(X, Y, ...) DMAX1(DX, DY, ...)	$Max(a_1, a_2, \dots)$	أكبر قيمة
مصحح مصحح حقيقي حقيقي دقة متضاعفة	MIN0(J, K, ...) MIN1(X, Y, ...) AMIN0(J, K, ...) AMIN1(X, Y, ...) DMIN1(DX, DY, ...)	$Min(a_1, a_2, \dots)$	أصغر قيمة
مصحح حقيقي	IDIM(J, K) DIM(X, Y)	$a_1 - \min(a_1, a_2)$	الفرق الموجب أو صفر ما بين (a_1, a_2)
حقيقي دقة متضاعفة مركب	SIN(X) DSIN(DX) CSIN(C)	$\sin(a)$	الدوال المثلثية لها اختصارات التقدير الدائري
حقيقي دقة متضاعفة مركب	COS(X) DCOS(DX) CCOS(C)	\cos	
حقيقي دقة متضاعفة	TAN(X) DTAN(DX)	$\tan(a)$	
مصحح حقيقي دقة متضاعفة	ISIGN(J, K) SIGN(X, Y) DSIGN(DX, DY)	$ a_1 \cdot \text{sgn}(a_2)$	نقل العلامة ما بين (a_1, a_2)
مصحح حقيقي مصحح	INT(X) AINT(X) IDINT(DX)	$\text{sgn}(a) \cdot [a]$	بتر المحدد للمصفوفة
مصحح حقيقي دقة متضاعفة	MOD(J, K) AMOD(X, Y) DMOD(DX, DY)	$a_1 - \text{sgn}(b)[b] \cdot a_2$, where $b = a_1/a_2$	إيجاد المثل ذي القيمة الصغرى المطلقة في قسم الباقي a_1 (modulo a_2)

ملحق (ب)

التمثيل الداخلى للبيانات

مقدمة

تتكون كل خلية ذاكرة من نبائط التوازن ، أى ، نبائط لها حالتى اتزان وتمثل هاتين الحالتين بالأرقام الثنائية 0 و 1 أى تسطيع كل نبيلة ثنائية التوازن تخزين بيت (*bit*) (اختصار رقم ثنائى *binary digit*) من المعلومات . وتستخدم عادة مجموعة أو سلسلة من الأرقام الثنائية لتمثيل وحدة من المعلومات .

حيث يمكن أن ترى سلاسل من الخانات الثنائية كأنها أرقام ثنائية ، إن تمثيل الأرقام فى النظام الثنائى هام فى علوم الحاسب . يبين الجدول ب - 1 التمثيل الثنائى لأول ستة عشر رقماً صحيحاً موجباً . يستخدم الدليل 2 أحياناً لتمييز الرقم الثنائى عن الرقم العشرى ، فمثلاً :

$(101101)_2$ تمثى الرقم الثنائى 101101

ذو التمثيل العشرى 45 . وتقع دراسة التحويل من الأرقام العشرية إلى الأرقام الثنائية وبالعكس خارج نطاق هذا الملحق .

جدول ب - 1

رقم عشري	رقم ثنائى
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000

كلمات وبيانات

تسمى الوحدة الأساسية للمعلومات التى تخزن أو تسترجع من خلية ذاكرة بكلمة (*word*) . يمكن أن تتكون الكلمة من بعض الأحجام الشائعة للكلمة 24 أو 30 أو 32 و 36 و 48 و 60 موضع ثنائى . هناك حاسبات بكلمات ذات طول ثابت وحاسبات بكلمات ذات طول متغير . ومع ذلك فكل كلمة فى الذاكرة يجب أن يكون لها عنوانها الخاص .

وعلاوة على ذلك ، كلما أدخلنا أو أخرجنا كلمة من الذاكرة ، فإن كل مواضعها الثنائية تنقل في آن واحد .

تسمح بعض الحاسبات أيضاً بتخزين أو استرجاع جزء من الكلمة من الذاكرة . تسمى هذا الجزء من الكلمة باسم بايت . وتتكون الباييت عادة من 6 أو 8 مواضع ثنائية ويمكن أن تأخذ عنواناً منفصلاً . يطلق على هذه الآلات بأن لها خصائص العنونة بالبايت (byte addressable) . وسلسلة CDC 6000—7000 بها كلمات تتكون من عشرة بايتات كل ذات ستة مواضع ثنائية وبهذا تتكون الكلمة من 60 موضعاً ثنائياً . لأغراض التوضيح ، سنعتبر أن آلتنا الحاسوبية تشبه سلسلة IBM 360/370 والتي لها كلمات مكونة من أربعة بايتات كل ذات 8 مواضع ثنائية . وبهذا تتكون الكلمة من 32 موضعاً ثنائياً كالتالي :

كلمة

Byte 0	Byte 1	Byte 2	Byte 3
--------	--------	--------	--------

وإن الكلمة كوحدة للمعلومات هامة أيضاً للآلات byte-addressable (المعنونة بالبايت) وتعطى عنوان الباييت أقصى اليسار .

بيانات حرفية

يستطيع أى مترجم فورتران عادة أن يتعرف على الحروف وعددها 48 وهي مذكورة في الفصل الثاني ، وبالتحديد 10 أرقام و26 حرفاً أبجدياً و 12 حرفاً خاصاً . حيث

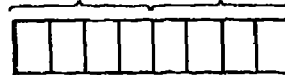
$$2^5 < 48 < 2^6$$

فيكون المطلوب على الأقل 6 مواضع ثنائية لتمثيل هذه الحروف . يمكن لأي كود حرف ذي 6 مواضع ثنائية أن يمثل حتى 64 حرفاً مختلفاً مما يسمح بعدد $2^6 = 64 - 36 = 28$ حرفاً خاصاً كحد أقصى . وتستخدم معظم حاسبات الجيل الثالث كود حروف ذات 8 — مواضع ثنائية وهذا يسمح بعدد يصل إلى 256 حرف وهو أكثر من كاف .

تكون عادة الحروف الأبجدية في ترتيبها الإ ، أى كود الحرف A أصغر من كود الحرف B ، وهكذا . يشكل هذا التكوين سلسلة مفروزة يمكن أن تستخدم . (انظر الفصل التاسع) .

يرتب عادة كود 8 موضع ثنائي (6 مواضع الثنائية أقصى اليسار ، وللقم الرقية كود الحرف المستخدم الآلات 370/90 II كود EBCDIC أى (نظام ترميز تمتد للتبديل العشري مكود ثنائياً) زبمة (الثلاثة) مواضع الثنائية أقصى اليمين . (انظر شكل ب - 1) . ويسمى

في منطقة



شكل ب - 1

(Extended Binary Coded Decimal Interchange Code) ويبين جدول ب - 2 الكود EBCDIC للحروف الأبجدية (62) والأرقام (10) لاحظ أن الكود بالنسبة لأي رقم يحتوي 1111 في قسم المناطق وتمثيلة الثنائي في القسم الرقي . ولما كان من غير الممكن تخزين أكثر من أربعة حروف ذات 8 — مواضع ثنائية في كلمة ذات 32 موضع ثنائي ، فإن هذا يفسر السبب في أن سعة الحرف في سلسلة IBM 360/4370 هي $M = 4$

جدول ب - ٢ أكواد EBCDIC

حرف	EBCDIC	حرف	EBCDIC	حرف	EBCDIC	حرف	EBCDIC
	منطقة أرقام		منطقة أرقام		منطقة أرقام		منطقة أرقام
A	1100 0001	J	1101 0001	S	1110 0010	0	1111 0000
B	0010	K	0010	T	0011	1	0001
C	0011	L	0011	U	0100	2	0010
D	0100	M	0100	V	0101	3	0011
E	0101	N	0101	W	0110	4	0100
F	0110	O	0110	X	0111	5	0101
G	0111	P	0111	Y	1000	6	0110
H	1000	Q	1000	Z	1110 1001	7	0111
I	1100 1001	R	1101 1001			8	1000
						9	1111 1001

بيانات رقمية

هناك مرتبتان، مختلفتان، لتكويد بيانات رقمية. أما تكويد الكمية أو تكويد الأرقام المفردة، نتاقتش أولاً تكويد الكمية.

تكويد الكمية

يوجد الفورتزان كلا من حسابات صحيحة ونقطة عائمة للحساب ويختلف التمثيل الداخلي لهذين النوعين من البيانات الرقمية :

(١) أعداد صحيحة

الطريقة المباشرة لتمثيل عدد صحيح في خلية ذاكرة تكون عن طريق التمثيل الثنائي. بفرض أن خلايا الذاكرة ذات 32 موضع ثنائي، يترك الموضع الثنائي الأول للإشارة، تخزن الأعداد الصحيحة 327 و 327. كالتالي :

327	0	0	0	...	0	1	0	1	0	0	0	1	1	1
-327	1	0	0	...	0	1	0	1	0	0	0	1	1	1

↑ موضع ثنائي للإشارة

وبالتالي، فأكبر رقم يمكن أن يخزن في 32 خانة ثنائية هو

$$0(\underbrace{111\dots 1111})_2$$

31 ones

ويمثل (١ - 2³¹) ويترتب على أي محاولة لتخزين أعداد صحيحة أكبر من السعة حدوث طمع ومن المحتمل أن تفقد الأرقام المنعوية قصوى.

لأسباب حسابية تخزن الأرقام السالبة في الذاكرة باستخدام ما يسمى بالتمثيل المكمل للأساس ناقص واحد *radix-minus-one* أو المكمل للأساس (*radix*) سنناقش أولاً هذه التمثيلات في النظام العشري حيث يطلق عليهم المكمل التاسع (*nine's*) والمكمل العاشر (*ten's*) على الترتيب. نحصل على المكمل التاسع (*nine's*) لرقم عشري بطرح كل رقم عشري من 9 والمكمل العاشر (*ten's*) لرقم هو مكمله التاسع (*nine's*) مضافاً إليه واحد.

على سبيل المثال :

رقم عشري	320	123	327
مكمل التسعات (nine's)	679	876	672
مكمل العشرات (ten's)	680	877	673

تسمى المكملات المناظرة للأرقام الثنائية المكمل الأحادي (one's) والمكمل الثنائي (two's) على الترتيب . بالمثل ، نحصل على المكمل الأحادي (one's) لرقم ثنائي بطرح كل خانة ثنائية من 1 (أى ، تحويل كل 0 إلى 1 وكل 1 إلى 0) والمكمل الثنائي (two's) لرقم عشري هو مكمل الأحادي (one's) مضافاً إليه واحد . فمثلاً .

رقم ثنائي	1010 ⁿ	10111	11001
المكمل الاحادي (one's)	01011	01000	00110
المكمل الثنائي (two's)	01100	01001	00111

يسمح لنا تمثيل الأرقام السالبة بالمكمل بإجراء الطرح باستخدام الجمع . (وحيث أن الضرب هو جمع متكرر والقسمة هي طرح متكرر ، بذلك يمكن إجراء كل الحسابات الصحيحة باستخدام الجمع) . نوضح كيف يعمل هذا الحساب بمثال .

حساب عادي	نظام عشري يستخدم المكمل التاسع (nine's)	نظام ثنائي يستخدم المكمل الأحادي (one's)
705	705	00000...0001011000001
-327	+672	+11111...1111010111000
378	①377	①00000...0000101111001
	→1	00000...0000101111010
	378	

لاحظ أن 1 الموجود على أقصى اليسار يحدف ثم يضاف إلى المجموع ، كما هو موضح بالأسهم - يسمى هذا بالترحيل عبر النهاية (end-around carry) وتفاصيل هذا الحساب تقع خارج نطاق هذا الملحق ولذا نكتفي بهذا القدر من الشرح .

(ب) أرقام بنقطة عائمة (ذات علامة عشرية)

دائمًا الجمع والطرح والضرب والصرف جزءها العشري لن تخزن قيمته ، وعلاوة على ذلك هذه القيود هي السبب الأساسي لاستخدام بنقطة عائمة واحدة .

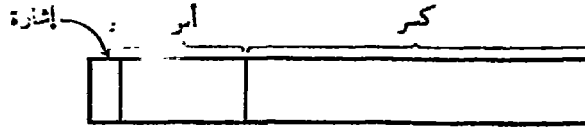
اعتبر الرقم 32.75 - يمكن أن يكتب

$$0.3275 \times 10^2 \text{ مع } 2 \text{ « (للأساس 10) »}$$

يمكن أن يستخدم تمثيل مثل - الثنائية . فمثلاً ، يمكن أن يكتب الرقم العشري 23.75 - كرقم ثنائي (100000.11)₂ ووضع في الشكل القياسي :

$$0.10000011 \times 2^6 \text{ مع } \left. \begin{array}{l} \text{إشارة} \\ \text{كسر} \\ \text{أس (أساس 2)} \end{array} \right\}$$

وضح أنه يمكن وضع كل رقم حقيقي في الشكل القياسي لنحصل على إشارة وحيدة ، كسر ثنائي وحيد ، وأس ثنائي وحيد ، بذلك يمكن أن يخزن داخلياً كما يلي :



في الكلمة ذات 32 موضع ثنائي ، نحتفظ بموضع ثنائي واحد للإشارة و ٧ مواضع ثنائية للأس ، ومن ثم ، عدد $24 = 32 - 8$ موضع ثنائي للكسر الثنائي . يجدر ملاحظة أن أي رقم حقيقي ، سواء غزن باستخدام حقل F له نفس التمثيل الداخلي كما هو موضح عاليه .

وحيث أن الأس يمكن أن يكون سالبا ، ويجب أن تشمل بعض التنظيمات لإشارته بداخل الموضع الثنائي ذاته ، فإن موضع ثنائي بدلا من استخدام موضع ثنائي واحد لتعريف إشارة الأس . ويتم ذلك بتحويل الأس الحقيقي إلى ما يسمى بـ (characteristic) $64 = 2^6 = 100000$ وبعد ذلك يوزن المميز بدلا من الأس الحقيقي . وفيما يلي العلاقة بين الأس الحقيقي والمميز .

الأس الحقيقي	63	...	1	0	-1	...	-61	-62	-63	-64
المميز	127	...	65	64	63	...	3	2	1	0

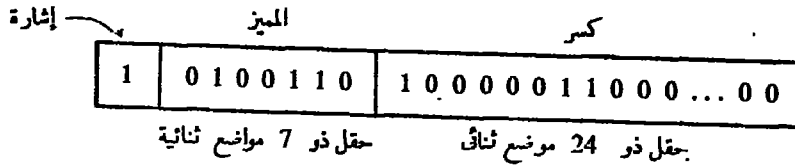
في حالة استخدام حقل ذي 8 مواضع ثنائية للأس ، نجمع $128 = 2^7$ إلى الأس الحقيقي لنحصل على المميز الخاص به . وهكذا) . مرة أخرى ، حيث تمثله الثنائي القياسي هو

$$-0.100001011 \times 2^6$$

الأس الحقيقي هنا هو ' تكون المميز هو

$$6 + 64 = 70 = (1001110)_2$$

وبذلك يوزن 32.75 — داخليا كما يلي:



نستطيع أن نرى أكبر رقم بالنقطة العائمة الذي يمكن أن يوزن في كلمة ذات 32 موضع ثنائي (باستخدام الشكل الثنائي ذو النقطة العائمة) هو 2^{63} وأصغر رقم هو 2^{-64} أي محاولة لتخزين أعداد أكبر من 2^{63} (أو أصغر من 2^{-64}) ستسبب في طفق في النقطة العائمة (أو قحط) . ويسبب حجم الكلمة المحدود تمثيلا داخليا غير دقيق للأرقام بالنقطة العائمة .

أخيرا ، تذكر أن في شكل النقطة العائمة ، لا يستخدم ترميز المكمل للكمور السالبة .

تكويد الخانات المفردة

وسيلة أخرى لتكويد البيانات الرقمية هي استخدام كود الحرف (الصيغة ذات المناطق) لكل خانة من خاناته . في مثل هذه الحالة تخصص الإشارة إلى حقل المنطقة (الخانة) على أقصى اليمين كما يلي :

منطقة	خانة	zone	digit	...	zone	digit	sign	digit
-------	------	------	-------	-----	------	-------	------	-------

في EBCDIC تمثل إشارة الزائد والنقص بـ 1100 و 1101 على الترتيب . كود الأرقام 327 و 327 — استخدام جدول ب - ٢ لا كواد الأرقام كما يلي :

	3		2	+	7		3		2	-	7	
	1111	0011	1111	0010	1100	0111	1111	0011	1111	0010	1101	0111

المذكور أعلاه هو الطريقة التي تمثل بها البيانات الرقمية في الإدخال والأخراج ومع ذلك فعندما تستخدم البيانات الرقمية في الحسابات ، فإنها تحول إلى شكل مختلف حتى يتم تشغيلها بسهولة . أحد هذه الأشكال هو النظام العشري المكون ثنائياً (BCD) (Binary Coded Decimal)

يشبه التكويد BCD للأرقام ، التكويد المذكور أعلاه فيما ، استخدام حقول المنطقة ويظهر كود الإشارة في الحقل أقصى اليمين . بمعنى آخر ، فإننا نستخدم تمثيل 4 ذي مواضع ثنائية لكل رقم عشري في العدد مع كود الإشارة في النهاية . بالتالي ، تكود 327 و 27- في BCD كما يلي :

	3		2	7	+		3		2	7	-
	0011	0010	0111	1100		0011	0010	0111	1101		

تم عملية الجمع في BCD بالجمع الثنائي للأرقام العشرية المفردة ، ومع ذلك ، عندما يتعدى المجموع تسعة ، يرسل وحدة إلى المكان التالي كما في الحساب العشري العادي . وأي ، مناقشة أخرى لهذا الحساب تكون خارج نطاق هذا الملحق .

المصطلحات العلمية (عربي - انجليزي)

(ا)

Alphanumeric	أبجدي رقمي
Echo check	اختبار الصدى
INPUT/OUTPUT	ادخال/اخراج
Format-free Input/Output	ادخال/اخراج بصياغة طليقة
Merging	الماج
Debugging	ازالة الخطا
Radix	اساس
Invoke	استدعاء
Name	اسم
Exponential	أسي
Machine	آلة
Storage locations	اماكن التخزين
Unconditional transfer	انتقال غير مشروط
Conditional transfer	انتقال مشروط
Selection	انتقاء

(ب)

Byte	بايت (مجموعة خانات)
Binary search	بحث ثنائي
Alternative	بديل
Subprograms	برامج فرعية
Red programs	برامج مخزونة
Source program	برنامج المصدر
Target program	برنامج الهدف
Card	بطاقة

Control card	بطاقة التحكم
Comment card	بطاقة تعليق
Sentinel card	بطاقة حارسه
Trailer card	بطاقة خلفية
Header card	بطاقة المقدمة
End-of-file card	بطاقة نهاية الملف
Dimension	بعث
Data	بيانات

(ت)

Key punching	تثقيب
Carriage control	تحكم العربيه
Conversion	تحويل
Storing numbers	تخزين الاعداد
Assignment	تخصيص
Hierarchy	تدرج هرمي
Binary coded decimal	ترميز عشري مكتوب بالرمز الثنائي
Left-justified	تضبيط جهة اليسار
Right-justified	تضبيط جهة اليمين
Parer 'asis-free expression	تعبير خال من الاقواس
Arithmetic expression	تعبير رياضي
Relational expression	تعبير مترابط
Logical expression	تعبير منطقي
Back-substitution	تعويض خلفي
Round-off	تقريب
Programming technique	تقنية البرمجة
Numerical integration	تكامل عددي
Graphing	تمثيل بياني
Internal representation	تمثيل داخلي
Distribution, frequency	توزيع ، تردد (تكرار)
Interval halving	تقسيم الفترات

(ث)

Constant	ثابت
Binary	ثنائي

(ج)

Precedence table	جدول الاسبقيات
Roots of equation	جذور المعادلة
Inventory	جرد
Mantissa	جزء عشري
Statement	جملة
Device	جهاز (وحدة)

(ح)

Computer	حاسب
Character/literal	حرف / حرفي
Package	حزمة
Arithmetic	حساب
Field	حقل
Loop	حلقة

(خ)

Blank	حالية
Exit from	خروج من
Flowchart	خريطة سير العمليات
Error	خطأ
Increment	خطوة التزايد
Linear	خطي
Argument	خلاصة
Algorithm	خوارزم (نظام حسابي)

	(د)	
Function		دالة
Subscript		دليل
	(ذ)	
Memory		ذاكرة
Two-dimensional		ذو بعدين
	(ر)	
Binary digit		رقمى ثنائى
Symbol		رمز
	(ز)	
Dummy		زائف
	(س)	
Record		سجل
Capacity		سعة
String		سلسلة من الحروف
	(ش)	
Form		شكل
	(ص)	
Integer		صحيح
Format		صيغة
Execution-time format		صيغة زمن التنفيذ
	(ض)	
Implied		ضمنى

(ط)

Method	طريقة
--------	-------

(ع)

Number	عدد
Prime number	عدد اولى
Real number	عدد حقيقى
Width	عرض
Decimal	عشرى
Operations	عمليات
Column	عمود
Element	عنصر
Return	عودة

(غ)

Unexecutable	غير قابلة للتنفيذ
Unformatted	غير مصاغ

(ف)

Exchange sort	فرز تبادلى
Internal sort	فرز داخلى
Bubble sort	فرز فقاعى

(ق)

Executable	قابلة للتنفيذ
Decision	قرار
Read-in	قراءة داخلية
Initial value	قيمة ابتدائية
Test value	قيمة الاختبار
Limiting value	قيمة محدودة
End value	قيمة النهاية

(ك)

Block	كتلة
Efficiency	كفاءة
Word	كلمة
Scalar	قيمة غير متجهة
Code	كود

(ل)

Infinite	لا نهائية
Programming language	لغات برمجة
Assembly language	لغة التجميع
High level language	لغة رفيعة المستوى

(م)

Sequential	متتابع
Nested	متداخلة
Relational	مترايط
Compiler	مترجم
Multidimensional	متعدد الأبعاد
Multiple	متعددة
Sequence	تسلسل
Double precision	متضاعف الدقة
Variable	متغير
Stream	مجرى
Assembler	مجمع
Set	مجموعة
Deck of cards	مجموعة بطاقات
Array	مجموعة مترابطة
Abnormal exit	خروج (خروج) غير طبيعي
Entry	مدخل
Complex	مركب
Conditional	مشروط

Operator	مشغل
Formatted	مصاغ
Matrix	مصفوفة
Equation	معادلة
Binomial coefficient	معامل ذو حددين
Generalized	معممة
Labeled	معنونة
Scale, factor	مقياس ، معامل
Complement	مكمل
Characteristic	خاصية
Logical	منطقي
Single precision	منفرد الدقة
Specifications	مواصفات

(ن)

Fixed point	نقطة ثابتة
Floating point	نقطة طليقة (عائمة)
Transfer of control	نقل التحكم
Mode of operation	نمط التشغيل
Mixed mode	نمط مختلط

(هـ)

Destructive	هدامة
Structure	هيكل

(و)

Unit	وحدة
Input unit	وحدة الادخال
Control unit	وحدة التحكم
Central processing unit (CPU)	وحدة التشغيل المركزية

المصطلحات العلمية (انجلىزى - عربى)

(A)

Abnormal	غير طبيعى
Address	عنوان
Algorithm	خوارزم (نظام حسابى)
Alphanumeric	ابجدى رقمى
Alternative	بديل
Argument	خلاصة
Arithmetic expression	تعبير رياضى
Array	مجموعة مترابطة
Assembler	مجمع
Assembly language	لغة التجميع
Assignment	تخصيص
Augmented	مزايدة

(B)

Back substitution	تعويض خلفى
Binary	ثنائى
Binary coded decimal (BCD)	ترميز عشرى مكتوب بالرمز الثنائى
Binary digit	رقم ثنائى
Binomial	ذو حددين
Blank	خالية
Block	كتلة
Bubble sort	مرز فقاعى
Byte	بايت (مجموعة خانات)

(C)

Capacity	سعة
----------	-----

Card	بطاقة
Carriage control	تحكم العربىة
Central processing unit (CPU)	وحدة التشغيل المركزية
Character	حرف
Characteristic	خاصية
Code	كود
Column	عمود
Comment	تعليق
Compiler	مترجم
Complement	مكمل
Complex	مركب
Computer	حاسب
Conditional transfer	انتقال مشروط
Constant	ثابت
Control	تحكم
Control unit	وحدة التحكم
Conversion	تحويل

(D)

Data	بيانات
Debugging	ازالة الخطأ
Decimal	عشرى
Decision	قرار
Deck of cards	مجموعة بطاقات
Destructive	هدامة
Device	جهاز (وحدة)
Dimension	بعمد
Distribution	توزيع
Double precision	متضاعف الدقة
Dummy	زائف

(E)

Echo check	اختبار الصدى
Efficiency	كفاءة
Element	عنصر
End-of-file	نهاية الملف
End value	تيمية النهاية
Entry	مدخل
Equation	معادلة
Error	خطأ
Exchange	تبادل
Executable	قابلة للتنفيذ
Execution-time	زمن التنفيذ
Exit from	خروج من (خرج)
Exponential	اكتبي
Expression	تعبير

(F)

Factor	معامل
Field	حقل
File	ملف
Fixed point	نقطة ثابتة
Flouting point	نقطة عائمة (طليقة)
Flowchart	خريطة سير العمليات
Form	شكل
Format	صيغة
Format-free	بدون صياغة
Frequency	تردد (تكرار)
Function	دالة

(G)

Generalized	معممة
Graphing	تمثيل بياني

(H)

Header card	بطاقة المقدمة
Hierarchy	تدرج هرمي
High level language	لغة رفيعة المستوى

(I)

Implied	ضمني
Increment	خطوة التزايد
Input-output	ادخال/اخراج
Integer	صحيح
Interval halving	سيف الفترات
Inventory	جرد - قائمة المخزون
Invoke	استدعاء

(K)

Keypunching	تثقيب
-------------	-------

(L)

Label	عنوان
Left-justified	مضببط جهة اليسار
Limiting value	قيمة محددة
Linear	خطي
Literal	حرفي
Logical	منطقي

(M)

Machine	آلة
Mantissa	جزء عشري
Matrix	مصفوفة
Memory	ذاكرة
Merging	ادماج
Method	طريقة

Mixed mode	نمط مختلط
Mode of Operation	نمط التشغيل
Multidimensional	متعدد الأبعاد
Multiple	متعدد (مضاعفات)

(N)

Name	اسم
Nested	متداخلة
Number	عدد
Numerical integration	تكامل عددي

(O)

Object program	برنامج الهدف
Operator	مشغل

(P)

Package	حزمة
Parenthesis-free	خال من الأقواس
Precedence table	جدول الأسبقيات
Prime number	سدد أولى
Programming languages	لغات برمجة
Programming technique	تقنية البرمجة

(R)

Radix	أساس
Record	سجل
Relational	مترابط
Return	عودة
Right-justified	مضببط جهة اليمين
Roots of equation	جذور المعادلة
Round-off	تقريب

(S)

Scale	مقياس
Scalar	كمية غير متجهة
Selection	انتقاء
Serial card	بطاقة حارسة
Sequence	تسلسل
Sequential	متتابع
Set	مجموعة
Single precision	منفرد الدقة
Source program specifications	برنامج المصدر مواصفات
Statement	جملة
Storage locations	اماكن التخزين
Stored programs	برامج مخزونة
Storing numbers	تخزين الاعداد
Stream	مجرى
String	سلسلة من الحروف
Structure	هيكل
Subprograms	برامج فرعية
Subscript	دليل
Symbol	رمز

(T)

Test	اختبار
Trailer card	بطاقة خلفية
Transfer of control	نقل التحكم
Two-dimensional	ذو بعدين

(U)

Unconditional	غير مشروط
---------------	-----------

Unexecutable		غير قابلة للتنفيذ
Unit		وحدة
	(V)	
Variable		متغير
	(W)	
Width		عرض
Word		كلمة

الفهرس الأبجدي

- برامج فرعية :
- ٢١٠ ، SUBROUTINE
- ٢٠١ ، FUNCTION
- ٢٠٩ ، متضاعف الدقة ،
- ٢١٤ ، مركب
- ٢١٠ . (SUBROUTINE) برنامج صغير فرعي
- ٢٠ ، برنامج المصدر ،
- ٢٠ ، برنامج الهدف ،
- ٢٠٧ ، بعد / متغير ،
- ٧ ، بطاقة
- ٩ ، بطاقة التحكم ،
- ٩ ، بطاقة تعليق ،
- ١١٥ ، بطاقة حارسة ،
- ١١٥ ، بطاقة خلفية ،
- ١١٥ ، بطاقة المقدمة ،
- ٩ ، بطاقة نهاية الملف ،
- ٩ ، بطاقة ID ،
- بيانات :
- ٣٥٣ ، تمثيل داخل ،
- ٩ ، FORTRAN (فورتران)
- ٩ ، مجموعة / بطاقات ،
- (DATA) بيانات
- ٢٧٨ ، ٢٥٣ ، جملة ،
- ٣٢٦ ، كلمة
- ٧ ، تثقيب
- ٥٩ ، تحكم العربة ،
- تدرن هرمي :
- لغات البرمجة ، ١٩
- العمليات ، ٣١
- تدرج هرمي للعملية ، ٢٦٥
- ترميز عشري مكتوب بالرمز الثنائي ، ٣٥٨
- تضييق جهة اليسار ، ٥٤
- أبجدي رقمي ، ٢٦ ،
- أبعاد متغيرة ، ٢٠٧ ،
- اختبار الصدى ، ١٦٩ ،
- إخراج
- جهاز (أو وحدة) ،
- مجرى ، ٦٠ ،
- مصاغ ، ٥٢ ،
- وحدة ، ١٩ ،
- إخطار فجائي ، ١٦٩ ،
- إدخال/إخراج
- غير مصاغ ، ٥١ ،
- مصاغ ، ٦٩ ، ٥٤ ، ٥٢ ،
- إدخال/إخراج بدون صيانة ، ٥١ ،
- إدماج ، ٢٢٩ ،
- إزالة الخطأ ، ١٦٩ ،
- أساس ، ٣٥٧ ،
- أساس ناقص واحد ، ٣٥٧ ،
- استدعاء برنامج فرعي ، ٢٠٣ ،
- اسم المتغير ، ٣٠ ، ١٩ ،
- أماكن التخزين ، ١٩ ،
- إنتقال غير مشروط ، ٣٧ ،
- إنتقال مشروط ، ٩٦ ،
- أو OR ، ٢٦٢ ،
- بايت ، ٣٥٣ ،
- بحث :
- متتابع ، ٢٣٠ ، ١٨٨ ،
- ثنائي ، ٢٣٠ ،
- خطي ، ١٩٥ ، ١٨٨ ،
- بحث ثنائي ، ٢٣٠ ،
- بحث خطي ، ١٩٥ ، ١٨٨ ،

- تفسيط جهة اليمين ، ٦٠ ، ٥٢ ،
تعبير :
- رياضي ، ٣١ ،
مترابط ، ٩٨ ،
منطقي ، ٢٦١ ،
تعبير خال من الأقواس ، ٣١ ،
تعبير مترابط ، ٩٨ ،
تعبير منطقي ، ٢٦١ ،
تقنية البرمجة ، ٢٢٥ ،
تكامل ، ٢٣٩ ،
تمثيل بياني ، ٢٨٨ ،
تمثيل داخلي ، ٣٥٤ ،
بيانات حرفية ، ٣٥٥ ،
بيانات عددية ، ٣٥٥ ،
تنصيف الفقرات ، ٢٢٤ ، ١٩١ ،
توزيع ، تردد (تكرار) ، ١٩١ ،
توزيع تكراري ، ١٩١ ،
ثابت :
- عدد حقيقي ، ٢٨ ،
عدد صحيح ، ٢٦ ،
النقطة الثابتة ، ٢٦ ،
النقطة الطليقة ، ٢٧ ،
خضاعف الدقة ، ٣٠٨ ،
مركب ، ٣١٢ ،
منطقي ، ٢٦١ ،
ثابت ذو نقطة ثابتة ، ٢٦ ،
ثابت ذو نقطة طليقة ، ٢٠ ،
جدول الأسقيات ، ٣١ ،
جنور المعادلة ، ٢٣٥ ،
جرد مخزون ، ١٩٧ ،
جزء عشري ، ٢٧ ، ٣٥٧ ،
جملة :
- تخصيص ، ٢٧ ،
حسابية ، ٢٦ ، ٢٧ ،
عددية ، ٩ ،
غير قابلة للتنفيذ ، ٢٠ ،
قابلة للتنفيذ ، ٢٠ ،
جملة التخصيص ، ٢٦ ، ١٠ ،
جملة غير قابلة للتنفيذ ، ٢٠ ،
جملة قابلة للتنفيذ ، ٢٠ ،
جملة قابلة للتنفيذ زائفة ، ١٣٥ ،
جملة النوع :
- ٣١٢ ، COMPLEX
٣٠٧ ، DOUBLE PRECISION
٢٨ ، INTEGER
٢٦١ ، LOGICAL
٢٨ ، REAL
جملة CALL ، ٢١٢ ،
جملة COMMON ، ٣١٧ ،
غير ممنونة (خالية) ، ٣١٧ ،
ممنونة ، ٣٢٠ ،
جملة DIMENSION ، ١٦٦ ،
جملة END ، ١٣ ، ٢٠ ،
جملة EQUIVALENCE ، ٢٢٢ ،
جملة FORMAT ، ٥٤ ،
جملة GO TO ، ١٥ ، ٩٤ ،
جملة IF :
- حسابية ، ١٠١ ،
منطقية ، ٩٩ ،
جملة IMPLICIT ، ٣٠٦ ،
جملة INTEGER ، ٢٨ ،
جملة PRINT ، ١١ ، ٥١ ،
جملة READ ، ٥٤ ، ٥١ ،
جملة RETURN ، ٢٢ ، ٣١٦ ،
جملة WRITE ، ٥٩ ،
حاسب البرامج المخزونة ، ١٩ ،
حرف :
- بيانات ، ٢٣٩ ، ٢٥٣ ،
سعة ، ٢٥٣ ،
مجموعة ، ٢٦ ،
حزمة :
- ٩ ، FORTRAN

- حساب :
- تعبير رياضي ، ٣١
- الجميل ، ٢٦
- جملة التخصيص ،
- جملة FUNCTION ، ٢٠٩ ، ٢٠٨
- عمليات حسابية ، ٣٠
- وحدة ، ١٨
- IF ، ١٠١
- حساب ذو نقطة ثابتة ، ٢٦
- حساب ذو نقطة طليقة ، ٣٠
- حساب من النوع :
- عدد حقيقي ، ٣٠
- عدد صحيح ، ٣٠
- متضاعف الدقة ، ٢٨
- مركب ، ٣١٣
- منفرد الدقة ، ٣٠٨
- حقل ، ٢٢٦ ، ٦٠ ، ٥٣
- حرفي ، ٦٥
- عرضي ، ٥٤
- مواصفات ، ٥٤
- موصف ، ٥٤
- حقل — A ، ٢٥٥
- حقل — D ، ٣٠٨
- حقل — E ، ٦٤ ، ٥٧
- حقل — F ، ٦٢ ، ٥٥
- حقل — G ، ٢٨١
- حقل — H ، ٢٥٩
- حقل — I ، ٦٢ ، ٥٤
- حقل — L ، ٢٦٤ ، ٦٥
- حقل — T ، ٢٨٠
- حقل — X ، ٦٢ ، ٥٩
- حقل حرفي ، ٦٥
- حقل — هو لوريث ، ٢٥٩
- حقل — D ، ٣٠٨
- حقيقي :
- التمثيل الداخلي ، ٣٥٥
- ثابت ، ٢٧
- حسابي / رياضي ، ٣١
- متنبر ، ٢٨
- حل المعادلة ، ٢٣٤
- حلقة تكرارية ، ١٠٢
- حلقة تكرارية لانهائية ، ١٠٦
- حاقة DO التكرارية ، ١٣٦
- المخرج من ، ١٤٣
- ضمنية ، ١٧٨
- متداخلة ، ١٤٨
- ممسة ، ٢٤١
- جملة DO ، ١٤٠ ، ١٣٦
- حاقة DO التكرارية الممسة ، ٣٤١
- حقة DO ضمنية ، ١٧٨
- حلقة DO الضمنية المتداخلة ، ١٨٠
- حلقة DO متداخلة ، ١٧٦ ، ١٤٨
- حلقة WHILE التكرارية ، ٣٣٨
- خروج (مخرج) طبيعي ، ١٤٣
- خريطة سير العمليات ، ٩٣
- خطأ :
- تحويل ، ٣٨
- تقريب ، ٣٨
- خطأ التحويل ، ٣٨
- خطأ التقريب ، ٣٩
- خطوط التزايد ، ١٤٠
- خلاصة ، ٢٠٤
- خوارزم ، ١١٠ ، ٩٣
- دالة أكبر عدد صحيح ، ٣٥٣
- دالة CMPLX ، ٢٦٩
- دليل ، ١٧٢ ، ١٦٤
- دوال مبنية داخلياً (مبينة) ، ٣٥١ ، ٣٤
- دوال المكتبة ، ٣٥١ ، ٣٤
- دوال FUNCTION :
- برنامج فرعي ، ٢٠١
- جملة التعريف ، ٢٠٢
- ذاكرة :
- عنصر ، ٢٨
- وحدة ، ١٩

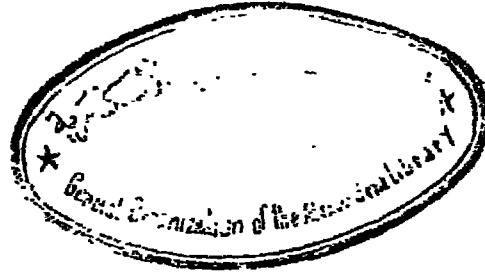
- رقم ثنائى ، ٣٥٣
رقم الجهاز (أو الوحدة) ، ٥٣
رمز التجميع ، ٢٤٥
رياضيات الحاسب ، ٣٨
- سجل ، ٢٢٦ ، ٦٦
سمة ، حرف ، ٢٥٣
سلسلة حرفية ، ٢٥٣
- شاغر (خال) ، ٢٦
شكل عشري ، ٢٧
- صحيفة تكويد ، ٧
صيفة أسية ، ٢٩
صيفة زمن التنفيذ ، ٢٨٦
- طريقة جاوس ، ٢٤٤
طريقة جاوس - جوردان ، ٢٥١
طريقة حوردان - جاوس ، ٢٥١
طريقة رايسون - نيوتن ، ٢٤٩
طريقة سيف ، ١٩٨
طريقة نيوتن - رايسون ، ٢٤٩
طريقة هورنر ، ٢٣٢
- عدد التكرار ، ٣٤٢
عدد أولى ، ١٠٤
عدد صحيح ؛
ثابت ، ٢٦
حساب ، ٠
تسمه ، ٣٠
متغير ، ٢٨
- عددى
ات ، ٣٥٧
- تكاملى ، ٢٣٧
ثابت ، ٢٦
عرض ، حقل ، ٥٣
عمود ، ١٧٢
عمود التكملة ، ٧
عنوان ، ١٩
- فرز :
إدماج ، ٢٢٩
إنتقاء ، ٢٢٧
تبادل ، ٢٢٧
داخلى ، ٢٢٦
فقاعى ، ٢٢٦
فرز تبادل ، ٢٢٧
فرز داخلى ، ٢٢٦
فرز بطريقة الدمج ، ٢٤٨
فرز الفقاعات ،
فرز معلومات حرفية ، ٢٥٨
فورتران هيكل ، ٢٣٤
- قرار ، ١٥
قراءة داخلية هدامة ، ٢٨
قراءة (من) غير هدامة ، ٢٨
قراءة (إلى) هدامة ، ٢٨
قيسة ابتدائية ، ١٤٠
قيسة الاختيار ، ١٤٠ ، ٣٤٢
قيسة محدودة ، ١٤٠
قيسة النهائية ، ١٤٠ ، ٣٤٠
- كتلة بيانات BLOCK DATA ، ٢٢٦
كتلة عامة غير ممنونة ، COMMON عالية ، ٣١٧
كتلة عبارة IF ، ٣٣٤
كتلة — ELSE ، ٣٣٦
كتلة — IF ، ٣٣٥
كفاءة ، ٢٢٥
كلمة ، ٣٥٢
كيفية غير متجهة ، ١٦٤ ، ٢٣٨
كود الصيغة ، ٥٤
- لغة :
آلة ، ١٩
تجميع ، ١٨
تدرج هرمى للغة ، ١٩
حاسب ، ١٨
مترجم ، ٢٠
مستوى رفيع ، ٢٠

- ٢٨٣ ، معامل ، مقياس ، نسبي ،
 ٩٨ ، معامل الترابط ،
 ٢٧٨ ، معامل التكرار ،
 ٢٠٤ ، معامل ذو حدين ،
 ٢٨٣ ، معامل القياس ،
 ٥٤ ، معامل P ،
 معاملات :
 ٢٩ ، رياضية ،
 ٩٨ ، مترابطة ،
 ٢٦٢ ، منطقية ،
 معاملات ، ٢٠٤ ، ٢٠٣ ،
 حلقة تكرارية ، ٣٤١ ، ١٤٠ ،
 معاملات الحلقة التكرارية ، ٣٤١ ، ١٤٠ ،
 ٢٢٩ ، مفتاح ،
 مكمل للتعبير المنطوق ، ٢٣٥ ،
 مكمل الواحد ، ٣٥٦ ،
 خاصة ، ٢٥٣ ،
 مواصفات :
 ٥٣ ، حقل ،
 ٥٤ ، I/O ،
 مواصفات حقل الإخراج :
 حقل — E ، ٦٤ ،
 حقل — F ، ٦٣ ،
 حقل — I ، ٦٢ ،
 حقل — X ، ٦٢ ،
 مواصفات حقل الإدخال :
 حقل — E ، ٥٧ ،
 حقل — F ، ٥٥ ،
 حقل — I ، ٥٥ ،
 حقل — X ، ٥٧ ،
 منطوق :
 ثابت ، ٢٦١ ،
 جملة التخصيص ، ٢٦٣ ،
 متغير ، ٢٦١ ،
 مشغل ، ٢٦٢ ،
 نظام معادلات مثلثي ، ٢٤٣ ،
 نقل التحكم :
- لغة الآلة ، ١٩ ،
 لغة التجميع ، ١٩ ،
 لغة رفيعة المستوى ، ٢٠ ،
 لغة المترجم ، ٢٠ ،
 مترجم ، ٢٠ ،
 متسلسلة فيبوناتشي ، ١٨٦ ، ١٩٦ ،
 متضاعف الدقة ، ٣٠٨ ،
 إدخال/إخراج ، ٣٠٨ ،
 برامج فرعية ، ٣١٠ ،
 ثابت ، ٣٠٨ ،
 حساب (رياضي) ، ٣٠٩ ،
 الدالة المكتتبية ، ٣٠٩ ،
 متغير : ٣٠٨ ،
 متعددة :
 سجلات ، ٦٦ ،
 عودة ، ٣١٥ ،
 مدخل ، ٣١٥ ،
 متغير ، ١٩ ،
 متغير ذو دليل ، ١٦٤ ،
 متغير زائف ، ٢٠٤ ،
 متغير مفهرس ، ١٣٦ ،
 متغيرات التحكم في الحلقة التكرارية ، ٢٢٨ ،
 جميع ، ١٩ ،
 مجموعة متراسة ، ١٦٤ ،
 مجموعة متراسة ذات بعد واحد ، ١٦٤ ،
 مجموعة متراسة ذات ثلاثة أبعاد ، ١٧٢ ،
 مجموعة متراسة متعددة الأبعاد ، ١٧٢ ،
 مخرج (خروج) ، غير طبيعي ، ١٤٣ ،
 مركب COMPLEX ، ٣١٢ ،
 عدد ، ٣١١ ،
 مصفوفات ، ٢٣٨ ،
 مصفوفة ، ١٧٢ ،
 مصفوفة مزادة ، ٢٤٥ ،
 مضبط جهة اليسار ، ٥٤ ،
 مضبط جهة اليمين ، ٥٣ ، ٥٥ ،
 معادلة ، ٢٢٤ ،
 معادلة خطية ، ٢٤٣ ،

٢٢٤ ، IF — هياكل
 ٢٤٠ ، FOR — هياكل
 ٢٢٨ ، WHILE — هياكل

٢٩٥ ، (AND) و
 وحدة التحكم ، ١١
 وحدة التشغيل المركزية ، ١٩
 وسط/ حسابي ، ١٨٤

داخل حلقة DO التكرارية ، ١٤٤
 غير مشروط ، ٩٤
 مشروط ، ٦٦
 ٢٢٦ ، IF ... THEN ... ELSE
 نمط 'تشغيل' ، ٣٠
 نمط مختلط ، ٢٢
 هياكل التحكم في الحلقة 'تكرارية' ، ٢٢٨



PROGRAMMING WITH FORTRAN

Including structured Fortran
Lipschutz

صدر أيضاً للناسخ في الحاسبات

BARTEE

* المدخل لعلم الحاسبات

GOTTFRIED

* البرمجة بلغة البيسك - شوم

LIPSCHUTZ

* البرمجة بالفورتران - شوم

LIPSCHUTZ

* الرياضيات الأساسية للحاسب - شوم

WOOLLARD

* المجزئات والحاسبات الدقيقة لطلبة الهندسة والفنيين

MORRIS

* الدوائر المتكاملة واستخدامات المجزئات الدقيقة

WOOLLARD

* الدوائر المتكاملة الرقمية والحاسبات

GOTTFRIED

* البرمجة بلغة البيسك - إنجليزي

الدار الدولية للنشر والتوزيع

الدار الدولية للنشر والتوزيع - مصر

الدار الدولية للنشر والتوزيع - لبنان

الدار الدولية للنشر والتوزيع - سوريا

الدار الدولية للنشر والتوزيع - ليبيا

الدار الدولية للنشر والتوزيع - تونس

الدار الدولية للنشر والتوزيع - الجزائر

الدار الدولية للنشر والتوزيع - المغرب

الدار الدولية للنشر والتوزيع - اليمن

الدار الدولية للنشر والتوزيع - العراق

الدار الدولية للنشر والتوزيع - الكويت

الدار الدولية للنشر والتوزيع - قطر

الدار الدولية للنشر والتوزيع - البحرين

الدار الدولية للنشر والتوزيع - عمان

الدار الدولية للنشر والتوزيع - سلطنة عمان