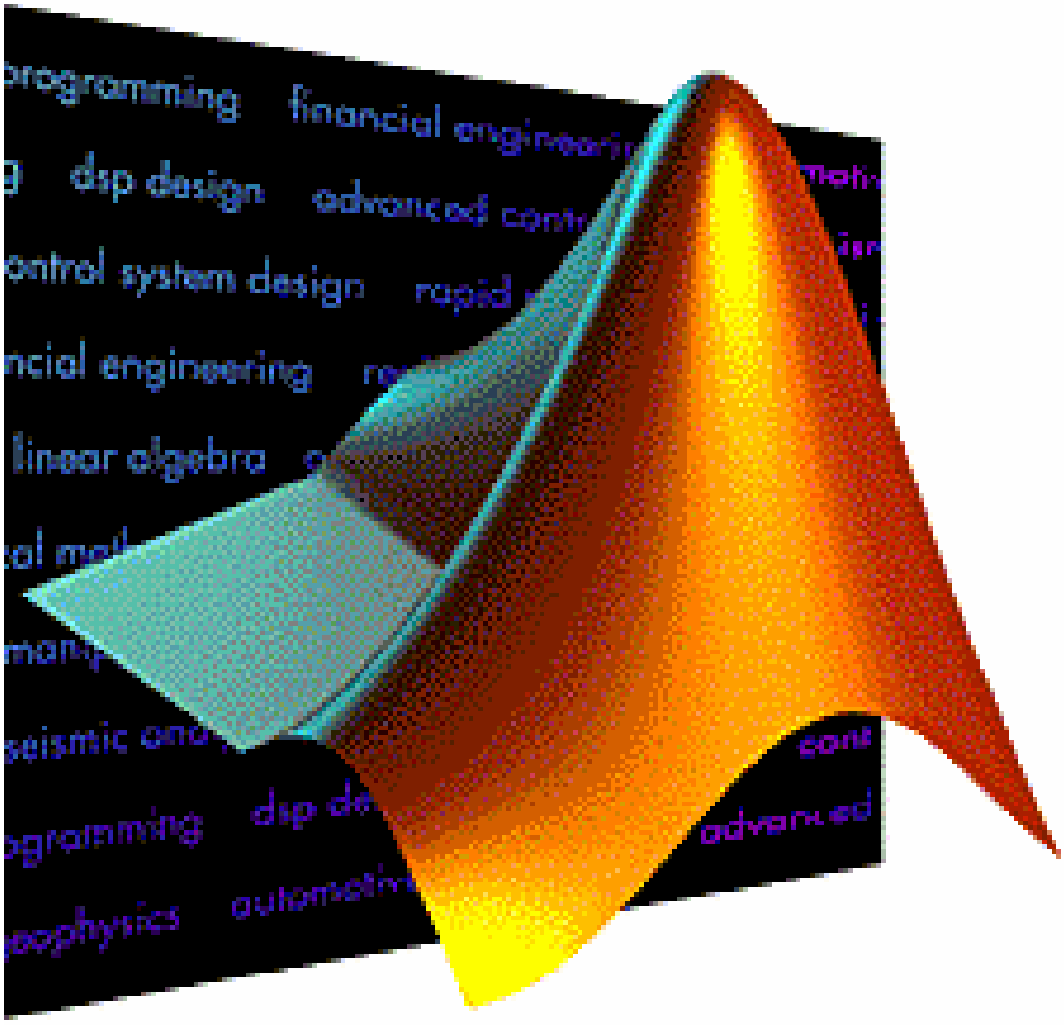
آخر تحديث لهذه النسخة: 2006/12/10

برمجة الرسم بلغة سي

باستخدام TURBO C++



تأليف: البراء عبد الرؤوف الرملي

albararamli@yahoo.com

المحتويات

حديث نبوي

عن ابن عباس قال سمعت رسول الله صلى الله عليه وسلم يقول:

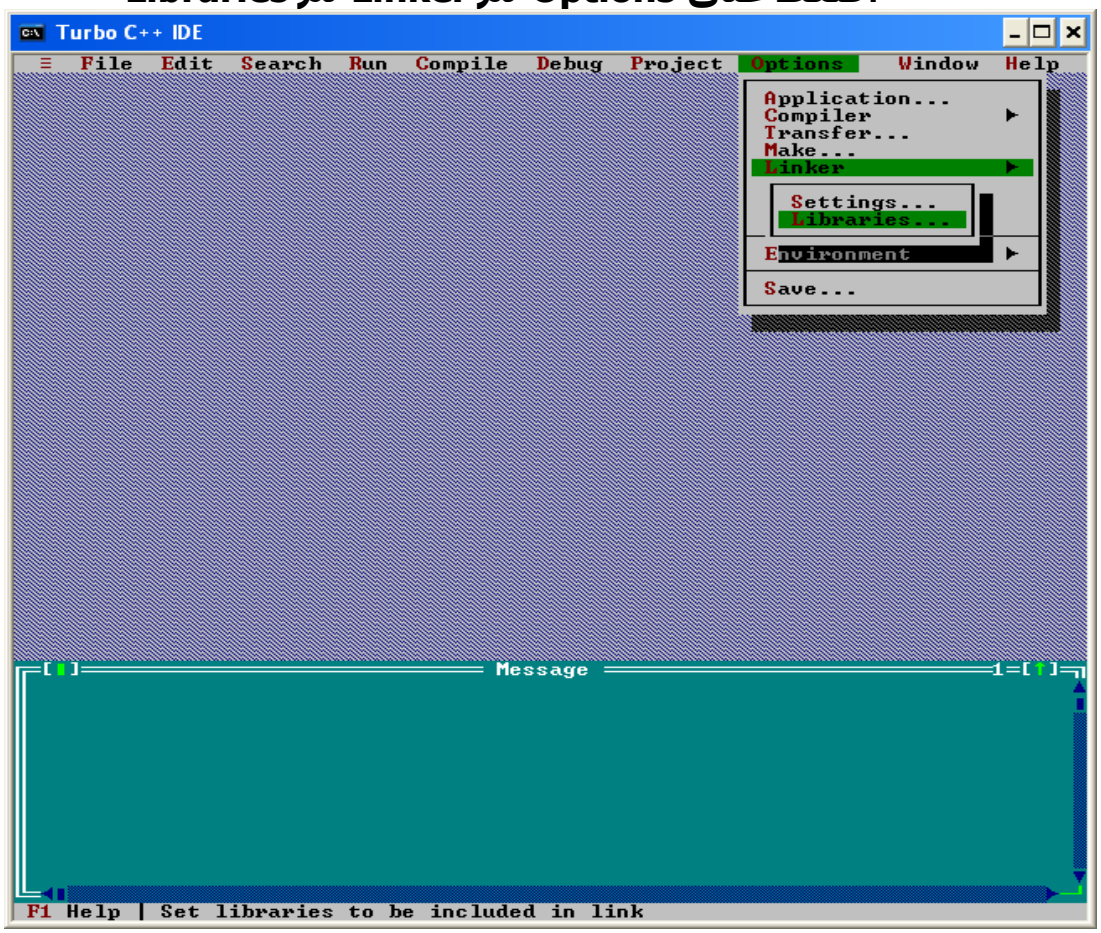
(كل مصور في النار يجعل له بكل صورة صورها نفسا فيعذبه في جهنم). - متفق عليه -

قال ابن عباس: فإن كنت لابد فاعلا فاصنع الشجر وما لا روح فيه.

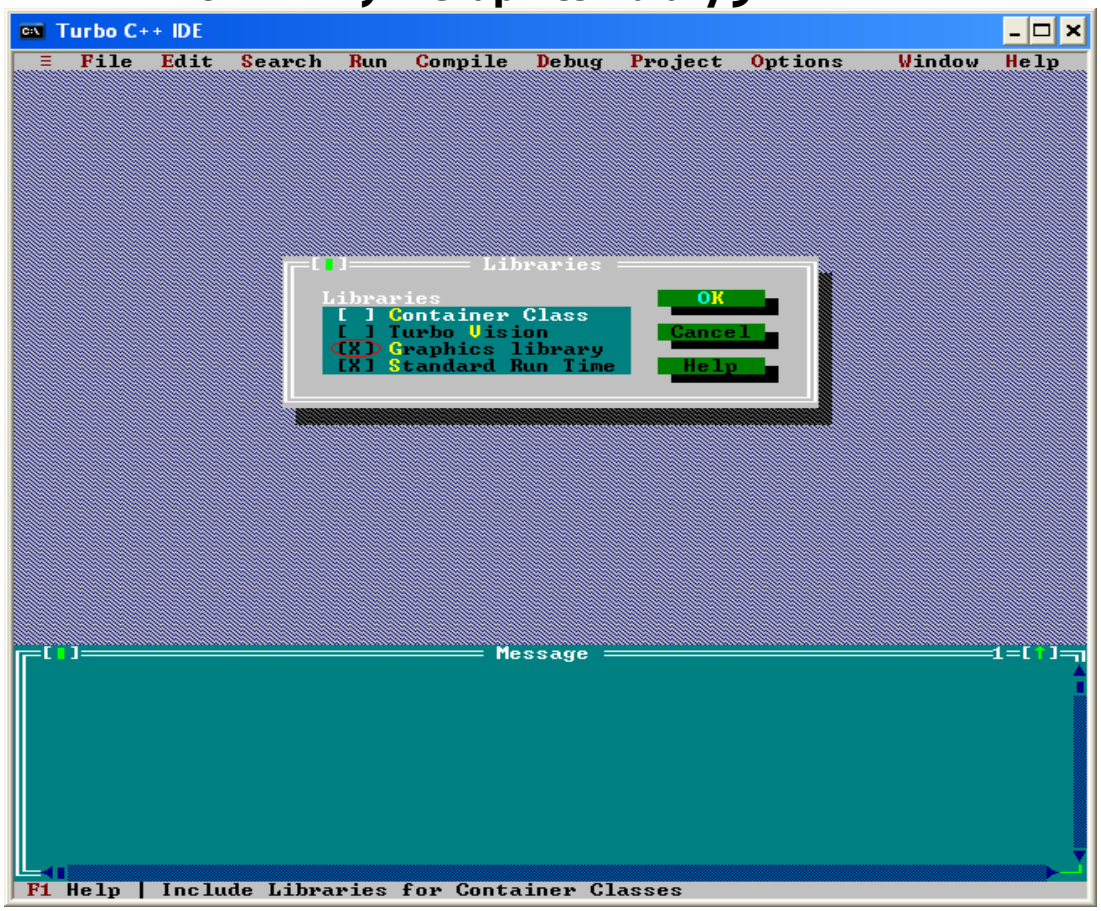
مفاهيم أساسية

إعداد TURBO C++ لبرامج الرسم

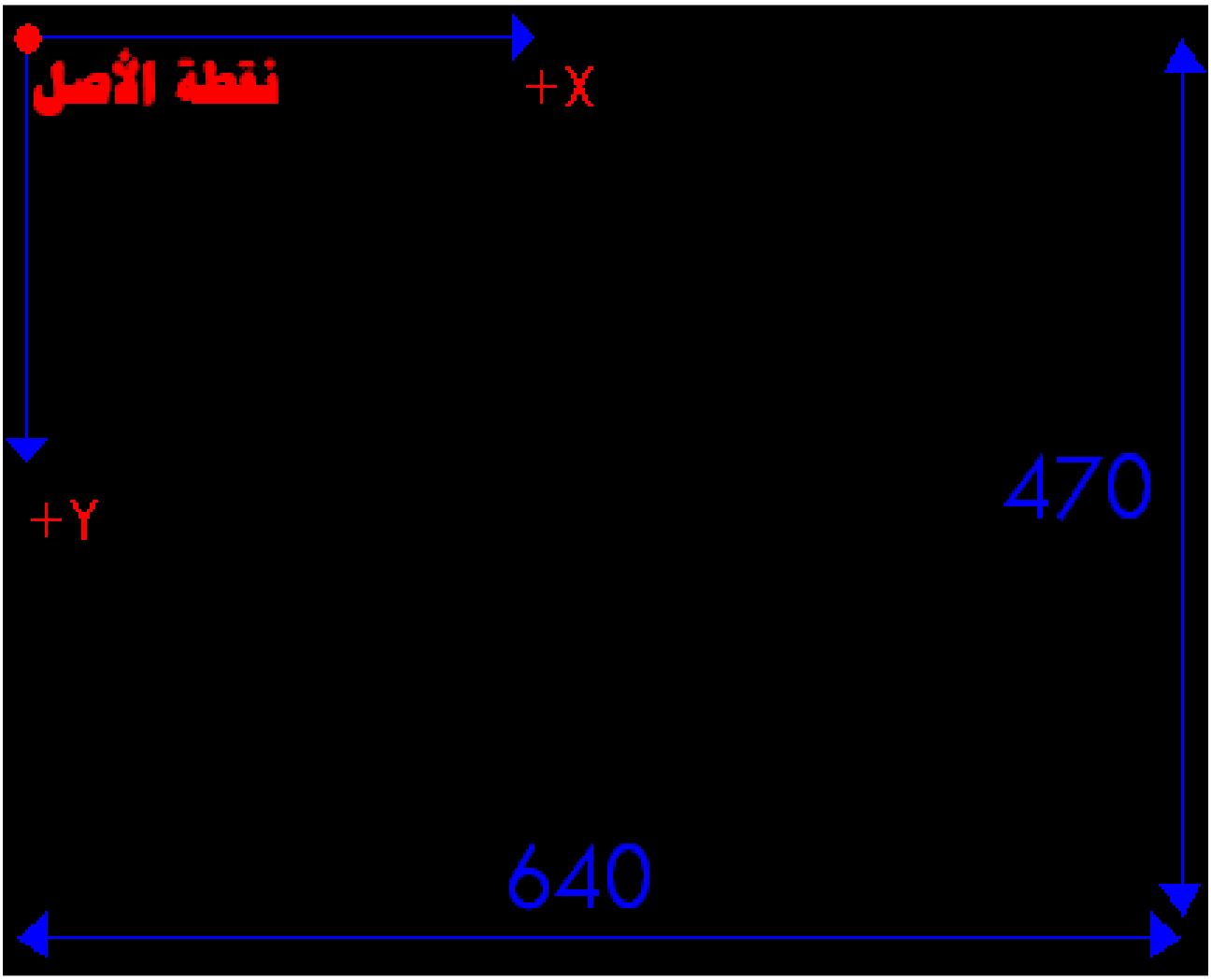
- اضغط على Options ثم Linker ثم Libraries



- اختر Graphics library ثم اضغط ok



شكل شاشة الرسم



يجب أن تعلم:

1. أن الشاشة قسمت أفقيا إلى 640 نقطة (pixel) ورأسيا إلى 470 نقطة (pixel)، وتكون دقة الصورة = 480 × 640
2. والـ (pixel) هي أصغر نقطة يمكن إضاءتها على الشاشة وهي مختصر للعبارة (picture cell).
3. نقطة الأصل تقع في الركن الأيسر أعلى الشاشة.
4. المحور الأفقي (محور X) يزداد من اليسار إلى اليمين.
5. المحور الرأسي (محور Y) يزداد من أعلى إلى أسفل.

ملاحظة:
تعمل البرامج مع كارت الفيديو VGA ، فإذا استخدمت كارت آخر فسوف يخرج الرسم عن حدود الشاشة.

الفصل الأول/ دوال الرسم في مكتبة

#include<graphics.h>

الشكل العام لبرامج الرسم

يجب كتابة النصوص الملونة بالأحمر في كل برنامج رسم:

#include<stdio.h> #include<conio.h>	
#include<graphics.h>	استدعاء مكتبة الرسم

void main()
{
int gdriver = DETECT, gmode, errorcode;
	تؤدي للتعرف على كارت الرسم الموجود

initgraph(&gdriver, &gmode, "c:\\tc\\bgi");	بمجرد استدعاء هذه الدالة يصبح الجهاز قادر على استقبال الأوامر من دوال الرسم المختلفة. تنبيه: يجب كتابة اسم الممر c:\\tc\\bgi على حسب مكان TURBO C++ في القرص الصلب.
.... (دوال الرسم تكتب هنا)	
getch(); }	تثبيت الصورة على الشاشة لحين الضغط على أي زر

قائمة بأسماء (دوال الرسم) الموجودة في مكتبة graphics.h وهي 83 دالة.

الدوال الموجودة في الكتاب مظللة باللون البنفسجي وهي 35 دالة

arc	imagesize
bar	initgraph
bar3d	installuserdriver
circle	installuserfont
cleardevice	line
clearviewport	linerel
closegraph	lineto
detectgraph	moverel
drawpoly	moveto
ellipse	outtext
fillellipse	outtextxy
fillpoly	pieslice
floodfill	putimage
getarccoords	putpixel
getaspectratio	rectangle
getbkcolor	registerbgidriver
getcolor	registerfarbgidriver
getdefaultpalette	registerbgifont
getdrivername	registerfarbgifont
getfillpattern	restorecrtmode
getfillsettings	sector
getgraphmode	setactivepage
getimage	setallpalette
getlinesettings	setaspectratio
getmaxcolor	setbkcolor
getmaxmode	setcolor
getmaxx	setfillpattern
getmaxy	setfillstyle
getmodename	setgraphbufsize
getmoderange	setgraphmode
getpalette	setlinestyle
getpalettesize	setpalette
getpixel	setrgbpalette
gettextsettings	settextjustify
getviewsettings	settextstyle
getx	setusercharsize
gety	setviewport
graphdefaults	setvisualpage
grapherrormsg	setwritemode
_graphfreemem	textheight
_graphgetmem	textwidth
graphresult	

بعض دوال الرسم العامة

دالة للخروج من نسق الرسم إلى نسق الكتابة/ closegraph

تؤدي لإغلاق نافذة الرسم والعودة إلى نافذة الكتابة العادية، وتكتب عادة بعد نهاية برنامج الرسم.

```
Closegraph();
```

وهي عكس الدالة ;("c:\\tc\\bgi", &gmode, &gdriver)initgraph
التى تؤدي لفتح نافذة الرسم.

دالة مسح الشاشة/ cleardevice

يمكنك مسح الشاشة بإدراج هذه الدالة:

```
cleardevice ();
```

وهي تقابل دالة ;()clrscr التى تؤدي إلى:
(مسح شاشة الكتابة العادية غير أن ;()clrscr لا تستخدم مع بيئة الرسم).

دالة للحصول على أقصى إحداثي سيني الشاشة /getmaxx

ترجع الدالة أقصى قيمة للإحداثي السيني على الشاشة، وهو يساوي 640.
تنبيه: ترجع الدالة رقم (640) وتخصصه للمتغير الصحيح :(m مثلا).

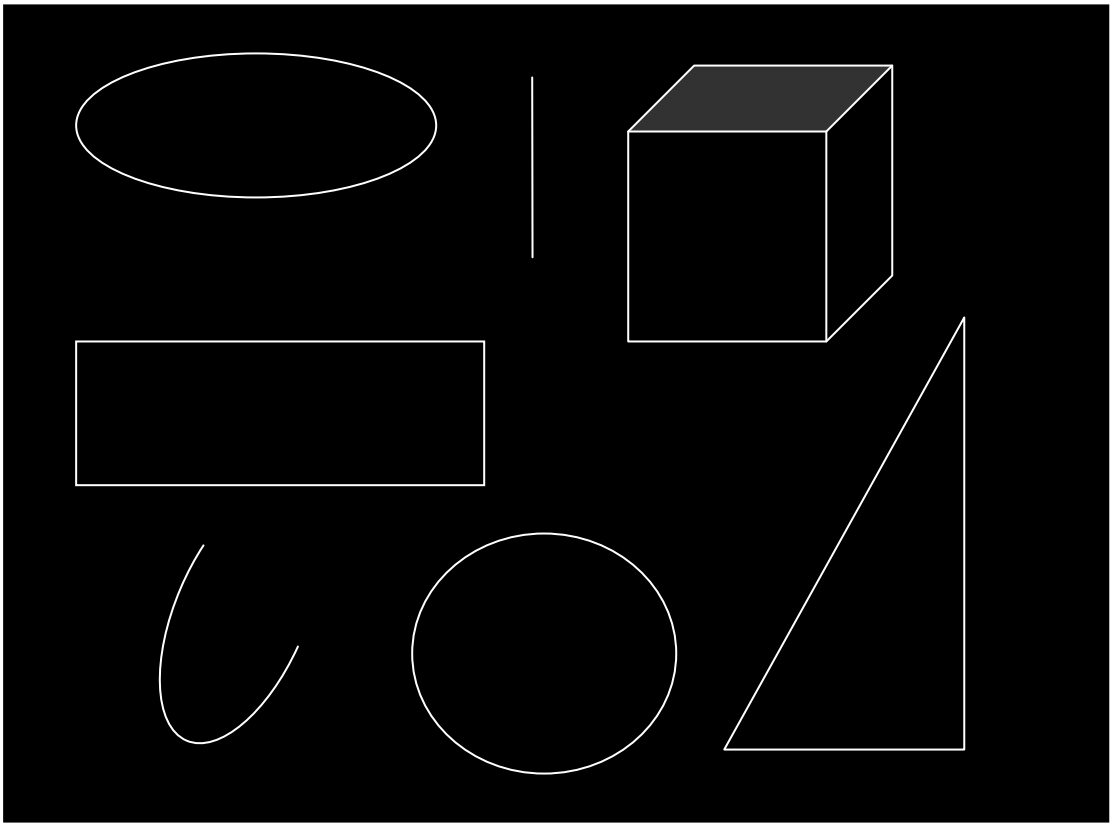
```
m=getmaxx();
```

دالة للحصول على أقصى إحداثي صادي للشاشة/ getmaxy

ترجع الدلة أقصى قيمة للإحداثي الصادي على الشاشة، وهو يساوي 470.
تنبيه: ترجع الدالة رقم (470) وتخصصه للمتغير الصحيح :(m مثلا).

```
m=getmaxy();
```

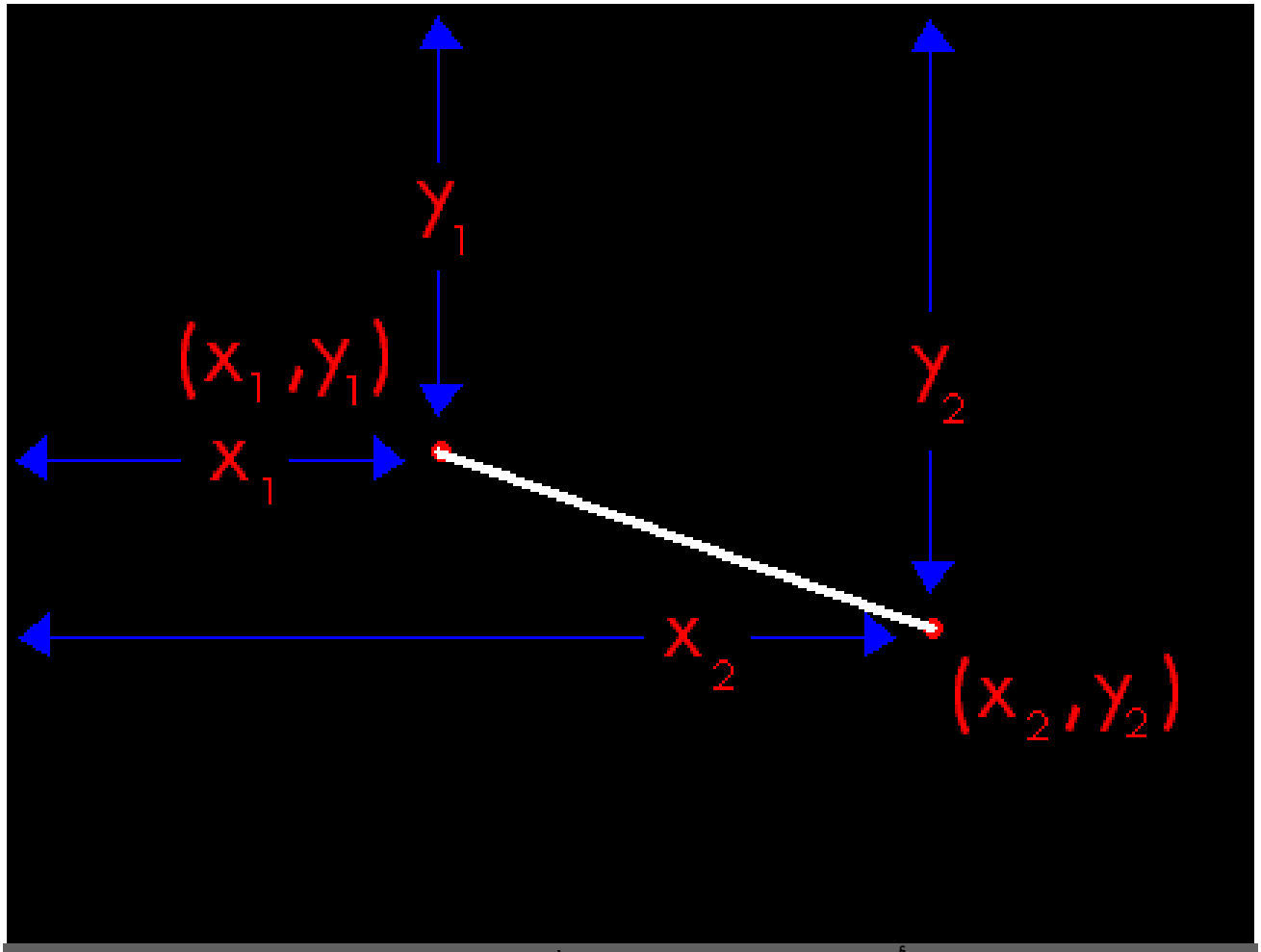
الفصل الثاني/الأشكال الهندسية



دالة رسم خط مستقيم/ line

تحتوي على 4 متغيرات عددية هم: إحداثي النقطة الأولى (x1,y1) وإحداثي النقطة الأخيرة (x2,y2).
ملاحظة: يمكن أن تكون الإحداثيات متغيرات صحيحة أو كسور عشرية.

Line(x1,y1,x2,y2);



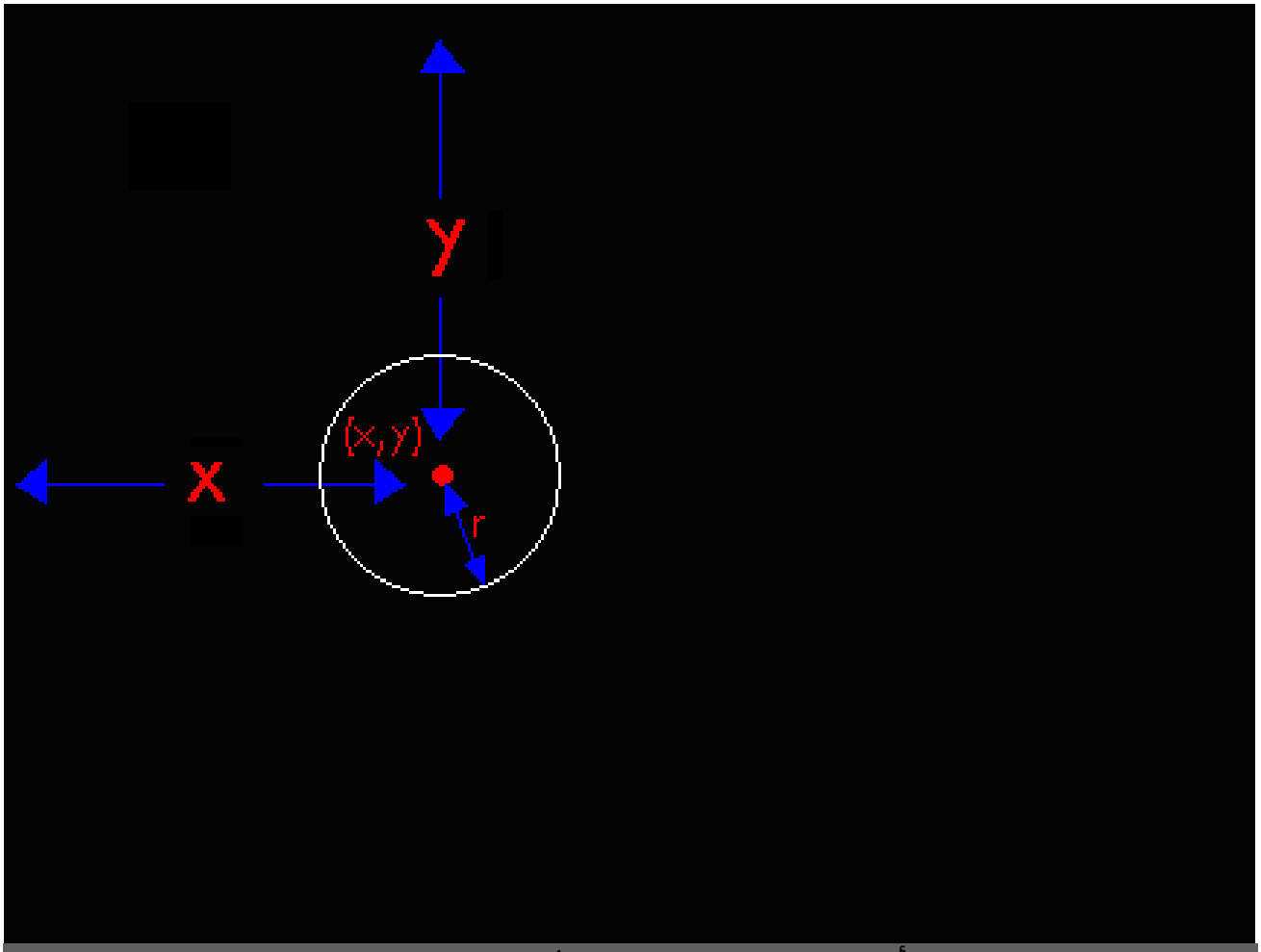
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int x1=50  ,y1=20;
int x2=200,y2=100;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
line(x1,y1,x2,y2);
getch();
}
```

دالة رسم دائرة/ circle

تحتوي على 3 متغيرات عددية هم: إحداثي المركز (x,y) ونصف القطر r.
ملاحظة: يمكن أن تكون x,y,z متغيرات صحيحة أو كسور عشرية.

circle(x,y,r);



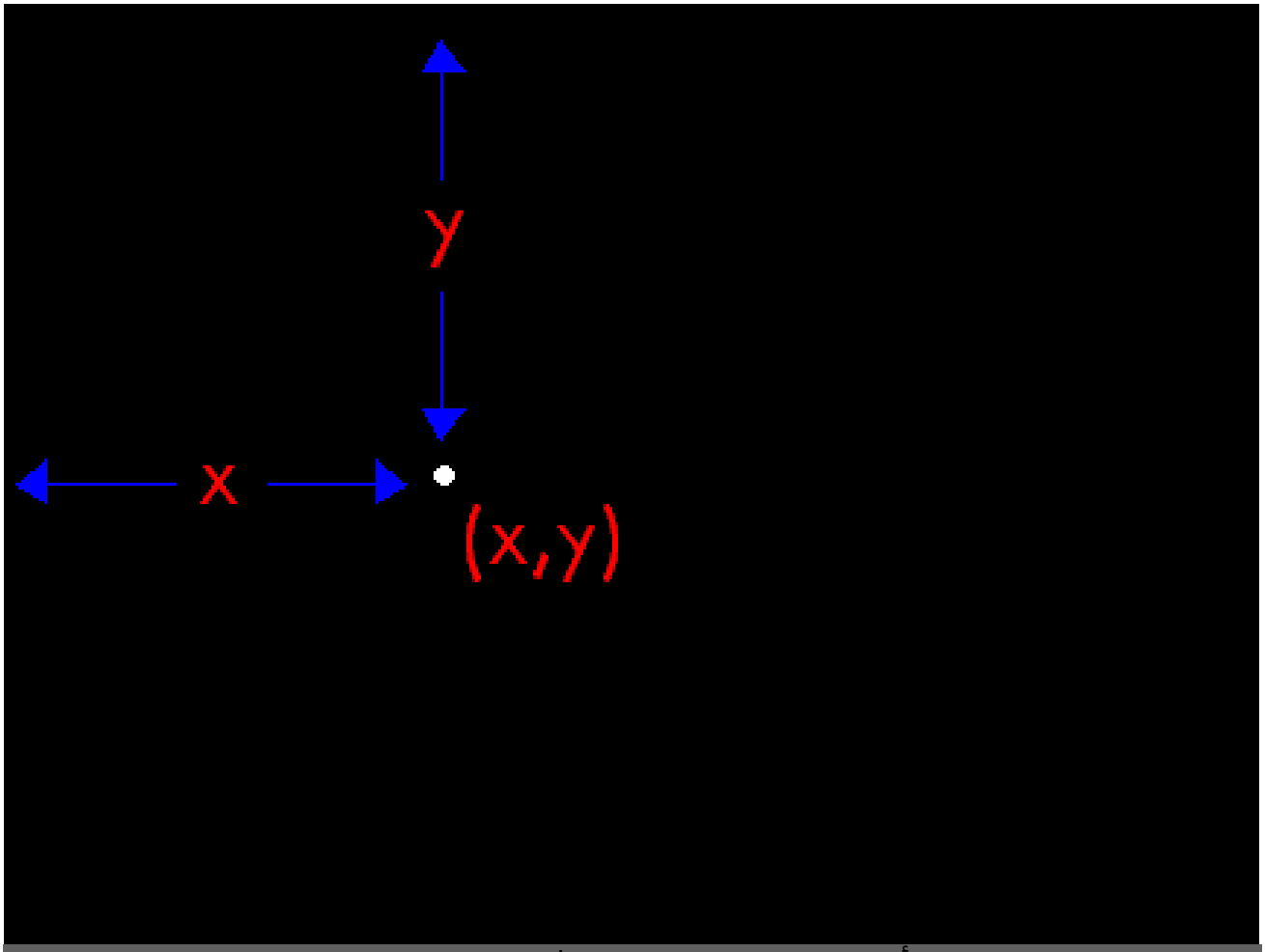
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int x=150,y=80,r=10;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
circle(x,y,r);
getch();
}
```

دالة رسم نقطة/ putpixel

تحتوي على 3 متغيرات عددية هم: إحداثي النقطة (x1,y1) ومتغير ثالث h هو لون النقطة، ضع رقم اللون في المكان المظلل، الألوان مرتبة من 0 إلى 15.
أما باقي الدوال فلا يوجد بها متغير للون لذا نستخدم معهم (دالة تغيير اللون).
ملاحظة: يمكن أن يكون (إحداثيات النقطة) متغيرات صحيحة أو كسور عشرية.

Putpixel(x,y,h);



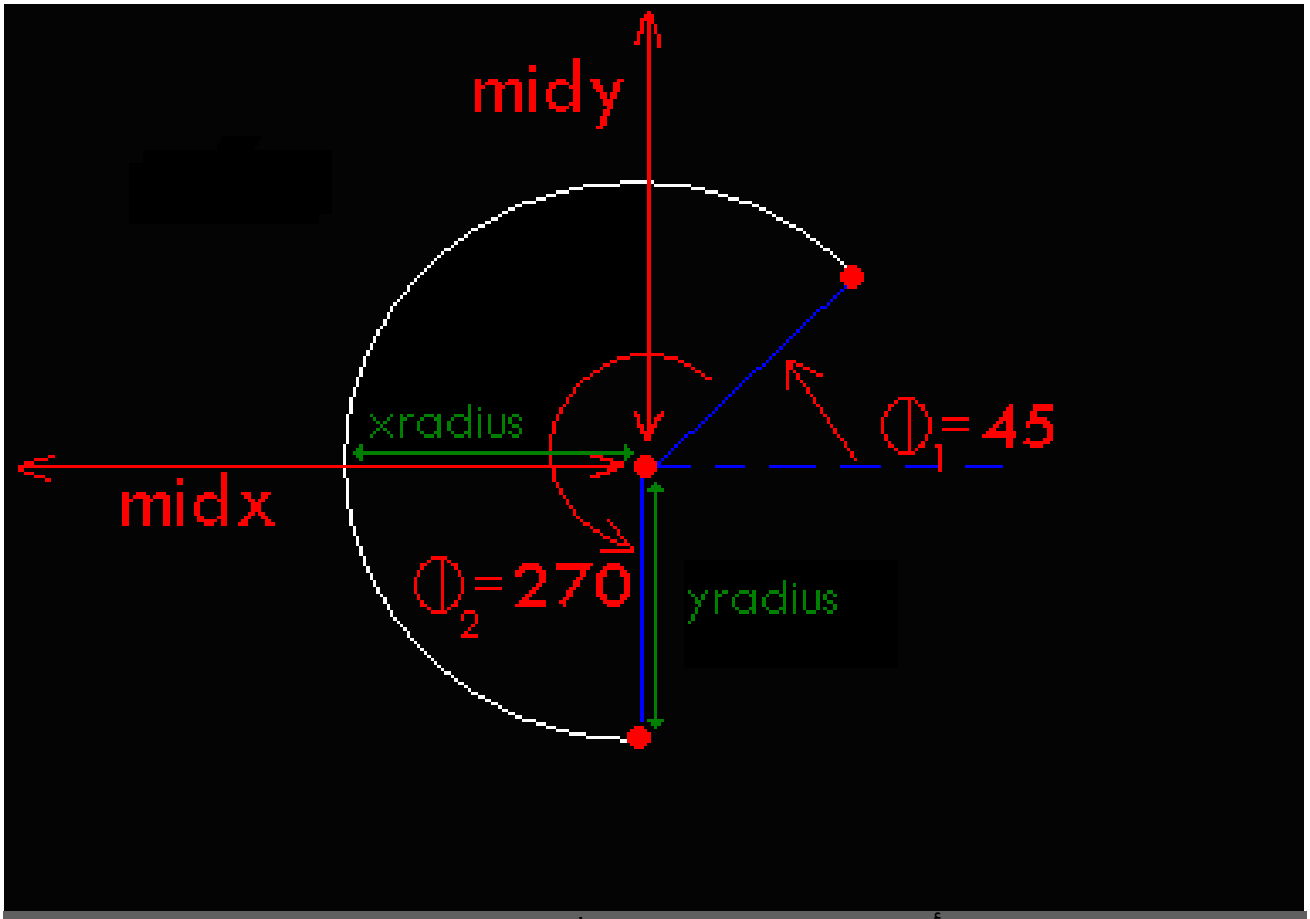
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
Int x=45,y=60,h=15;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
putpixel(x,y,h);
getch();
}
```

دالة رسم أقواس دائرية/ arc

إحداثي المركز=(midx, midy)
زاوية البدء (Φ1) =Stangle
زاوية النهاية (Φ2) =Endangle
نصف القطر=Radius

arc(midx, midy, stangle, endangle, radius);



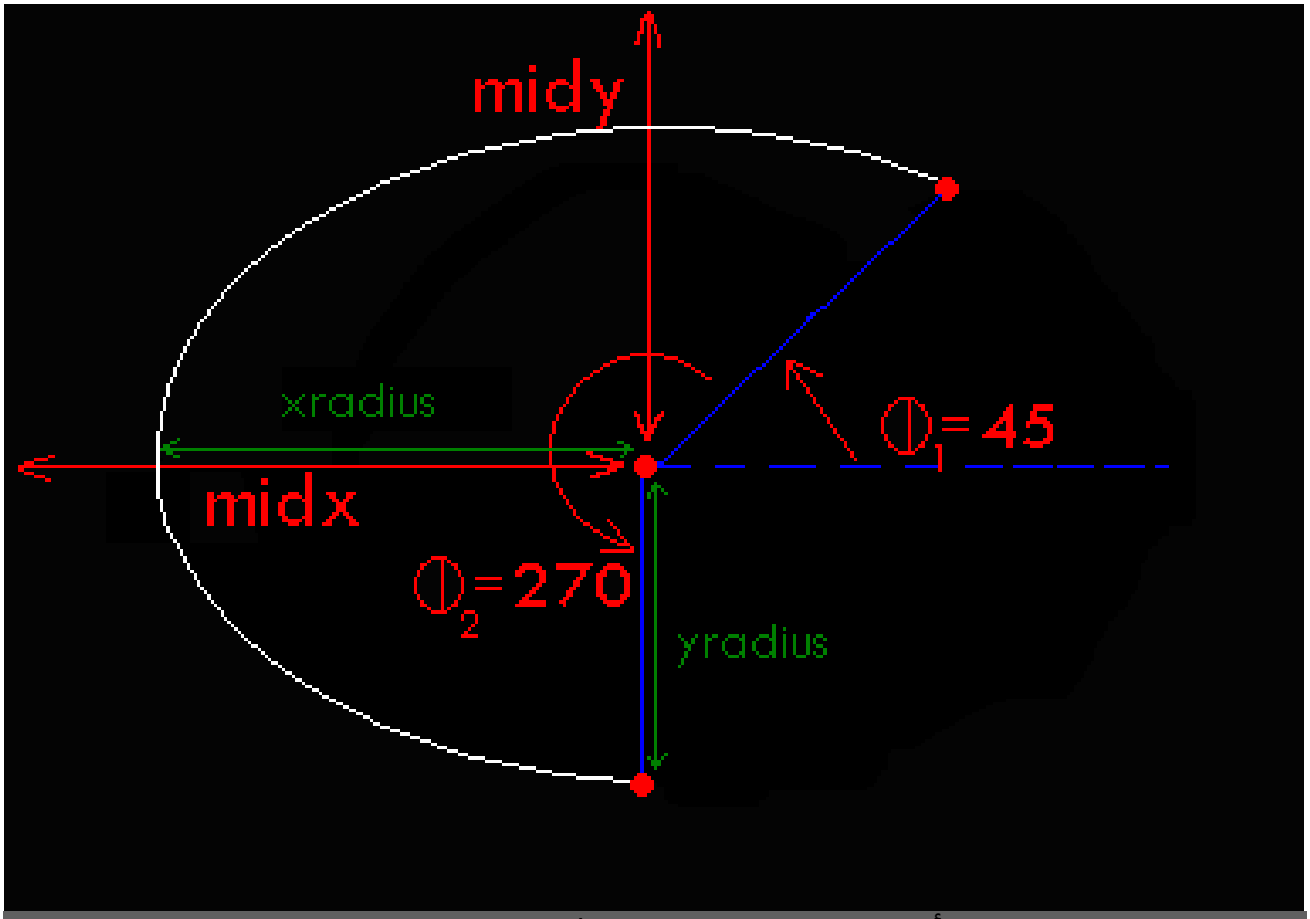
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
   int midx=320, midy=240, stangle = 45;
   int endangle = 135, radius = 100;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
arc(midx, midy, stangle, endangle, radius);
getch();
}
```

دالة رسم قطع ناقص/ ellipse

إحداثي المركز=(midx, midy)
زاوية البدء (Φ1) =Stangle
زاوية النهاية (Φ2) =Endangle
نصف قطر الإحداثي x =Xradius
نصف قطر الإحداثي y =Yradius

Ellipse(midx, midy, stangle, endangle, xradius, yradius);



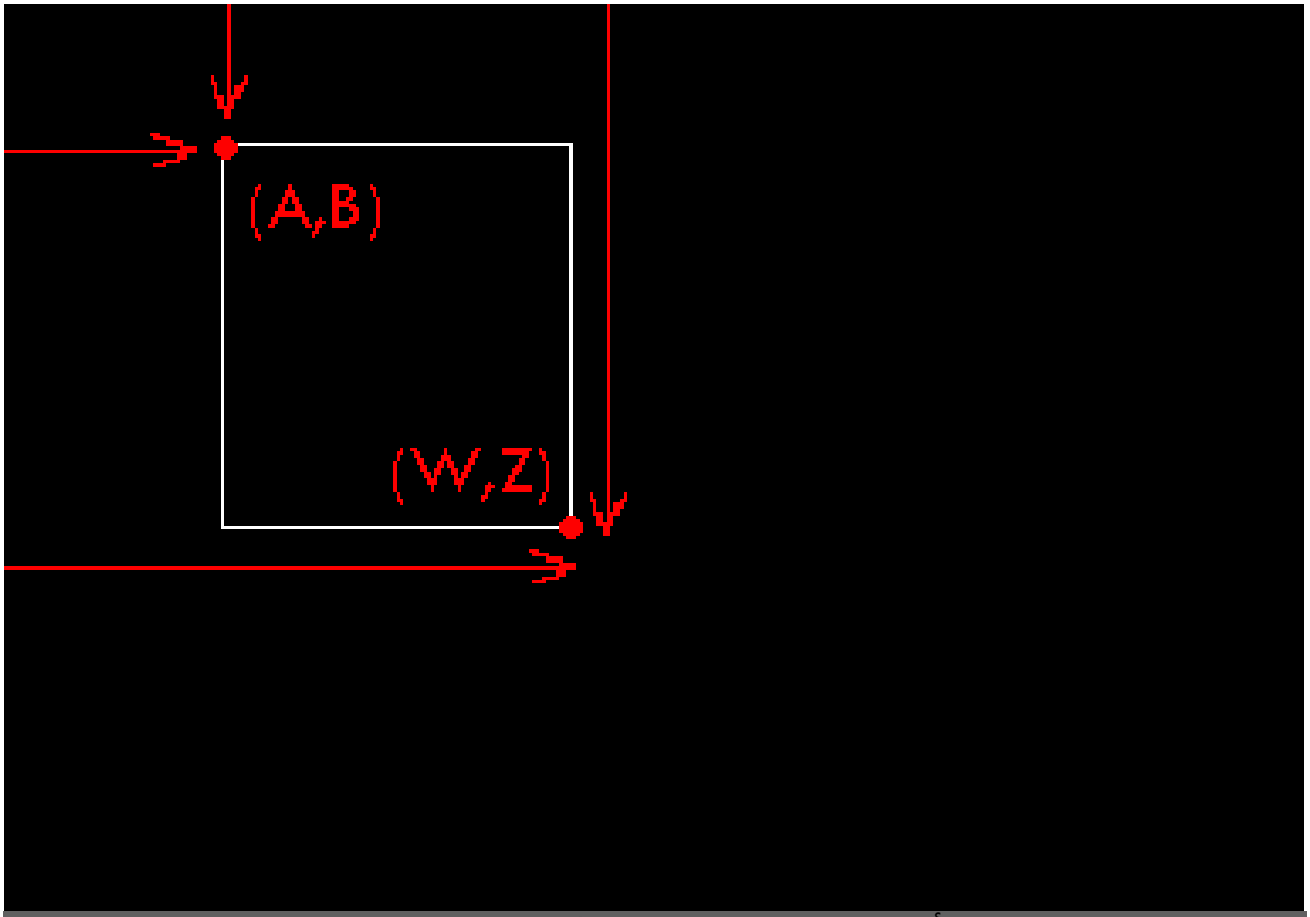
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
   int midx=320, midy=240, stangle = 45;
   int endangle = 135, radius = 100;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
ellipse (midx, midy, stangle, endangle,xradius,yradius);
getch();
}
```

دالة رسم مستطيل/ rectangle

إحداثي الركن الأيسر = (A,B)
إحداثي الركن الأيمن = (W,Z)

rectangle(A,B,W,Z);



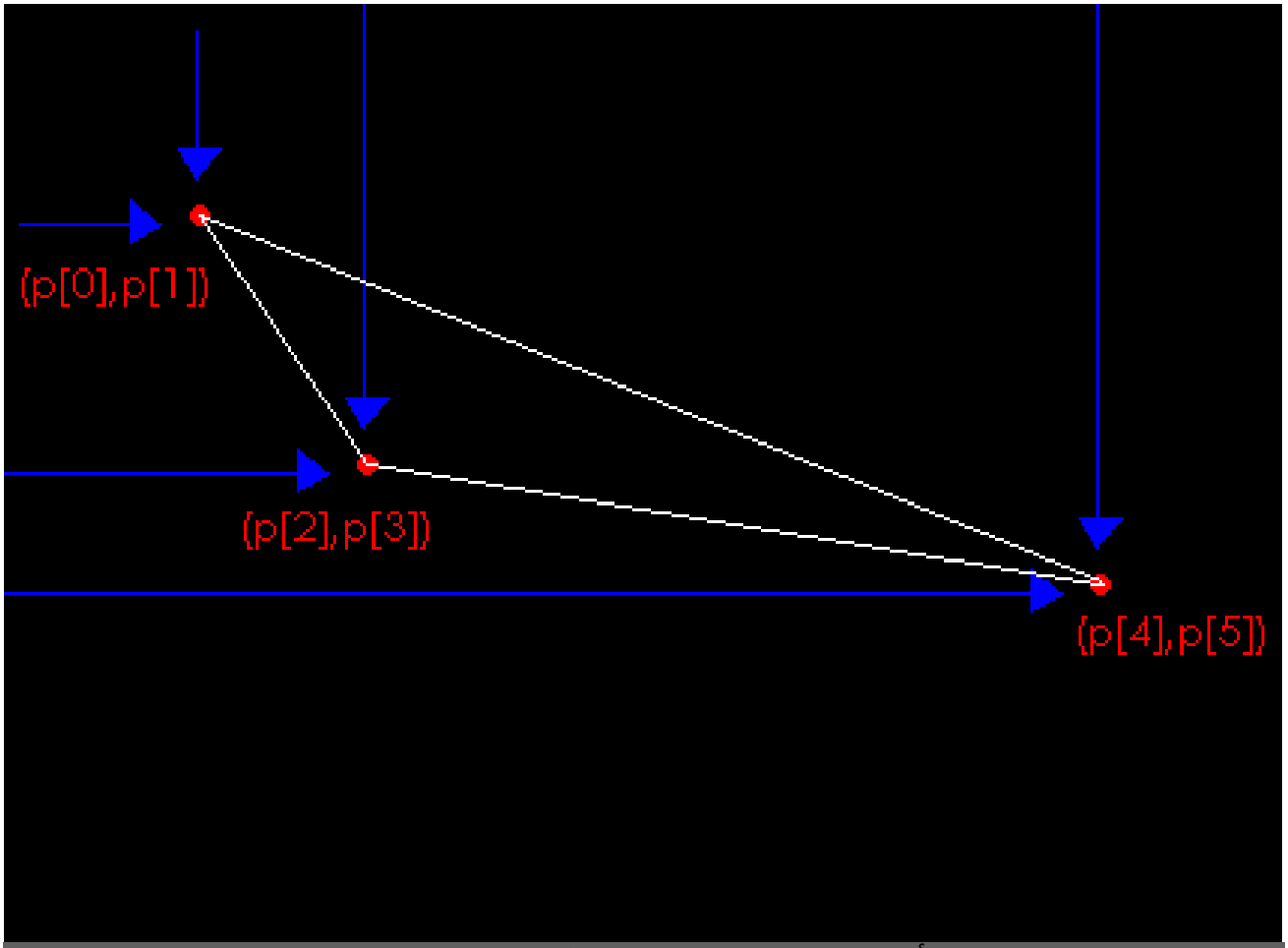
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
  Int a=10,b=20,w=150,z=200;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
rectangle(a,b,w,z);
getch();
}
```

دالة رسم الأشكال المضلعة/ drawpoly

اسم مصفوفة النقاط هي p[n] حيث n عدد النقاط.

Drawpoly(n,p);

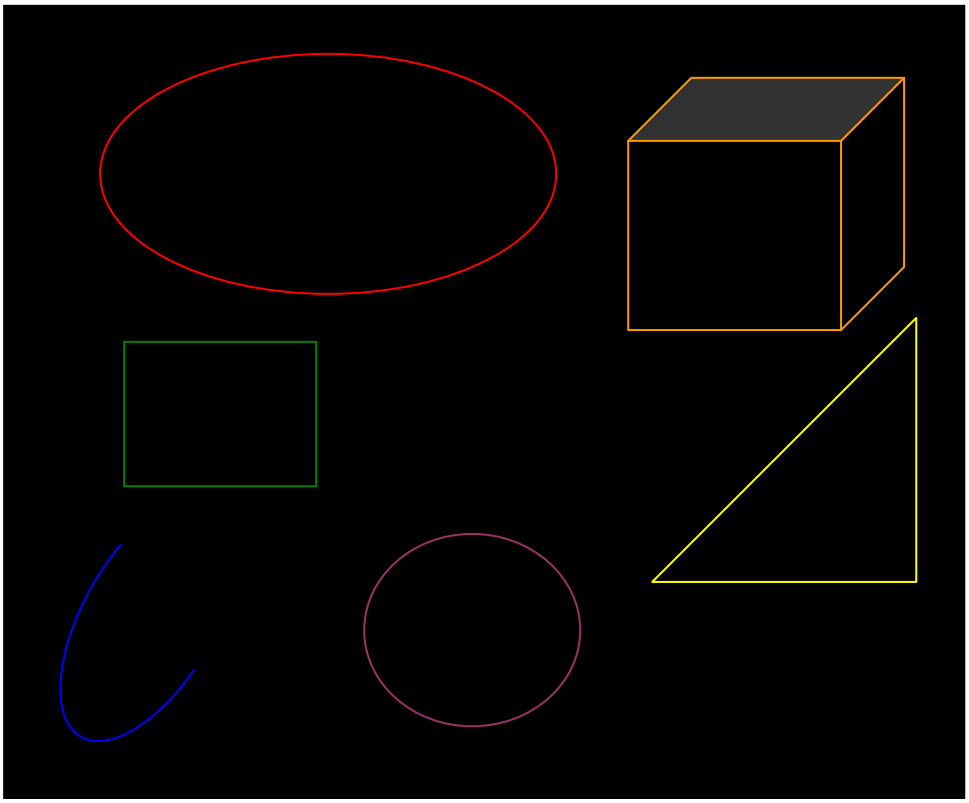


المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
Int p[3];
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
P[0]=10;p[1]=15;
P[2]=100;p[3]=200;
P[4]=200;p[5]=250;
fillpoly(3,p);
getch();
}
```

الفصل الثالث/

تلوين الأشكال الهندسية



دالة تغيير نوع خط الرسم / setlinestyle

يمكنك تغيير لون ونوع السطح الأشكال المرسومة حيث:

- **a عدد صحيح من 1 إلى 4 ويرمز لنوع الخط .**
- **b عدد صحيح من 0 إلى 12 ويرمز للون السطح.**
- **c عدد صحيح إما 1 خط عادي أو 3 خط سميك.**

```
Setlinestyle(a,b,c);
```

ملاحظة: يجب أن تكتب (هذه الدالة) قبل (دالة الرسم)، وإذا لم تستعمل دالة تغير اللون فإن لون السطح سيكون أبيض تلقائيا.

دالة تغير لون الرسم/ Setcolor

يمكنك تحديد لون الرسم باستخدام الدالة setcolor وذلك بإدراج رقم اللون بين قوسي الدالة في المكان المظلل:

```
setcolor( 15 );
```

0	1	2	3	4	5
أسود	أزرق	أخضر	كحلي	أحمر	بنفسجي

6	7	8	9	10	11
بني	رصاصي فاتح	رصاصي غامق	أزرق فاتح	أخضر فاتح	كحلي فاتح

12	13	14	15
أحمر فاتح	بنفسجي فاتح	أصفر	أبيض

ملاحظة: يجب أن تكتب (هذه الدالة) قبل (دالة الرسم)، وإذا لم تستعمل دالة تغير اللون فإن لون الرسم سيكون أبيض تلقائيا.

دالة تغير لون خلفية الشاشة/ setbkcolor

يمكنك تحديد لون الخلفية باستخدام الدالة setbkcolor وذلك بإدراج رقم اللون بين قوسي الدالة في المكان المظلل:

```
setbkcolor( 4 );
```

ملاحظة: يجب أن تكتب (هذه الدالة) قبل (دالة الرسم)، وإذا لم تستعمل دالة تغير اللون فإن لون الخلفية سيكون أسود تلقائيا.

دالة للحصول على لون الخلفية/ getbkcolor

تمكنك من الحصول على القيمة العددية للون الخلفية.
مثلا: لو كان لون الخلفية هو الأسود، فسوف ترجع الدالة رقم (0) وتخصصه للمتغير:(m مثلا).

```
m=getbkcolor();
```

دالة للحصول على لون خط الرسم/ getcolor

تمكنك من الحصول على القيمة العددية للون الرسم.
مثلا: لو كان لون الرسم هو الأحمر، فسوف ترجع الدالة رقم (5) وتخصصه للمتغير:(m مثلا).

```
m=getcolor();
```

دالة لتلوين الأشكال الهندسية/ floodfill

لصب اللون داخل شكل مغلق، حيث (G,F) إحداثيات نقطة تقع داخل الشكل المغلق.

```
floodfill(G,F,getmaxcolor());
```

تنبيه: يجب أن تكتب (هذه الدالة) بعد (دالة الرسم).
تستعمل هذه الدالة لتلوين الأشكال الهندسية المغلقة مثل: الدائرة والمثلث والمربع والمستطيل....
ويؤدي استعمالها مع غير الأشكال المغلقة إلى تلوين الشاشة بالكامل.

دالة للحصول على آخر لون في سلسلة الألوان/ getmaxcolor

تمكنك من الحصول على القيمة العددية لآخر لون، مثلا: لو كان لون الرسم هو الأسود، فسوف ترجع الدالة رقم (15) وتخصصه للمتغير:(m مثلا).

```
m=getmaxcolor();
```

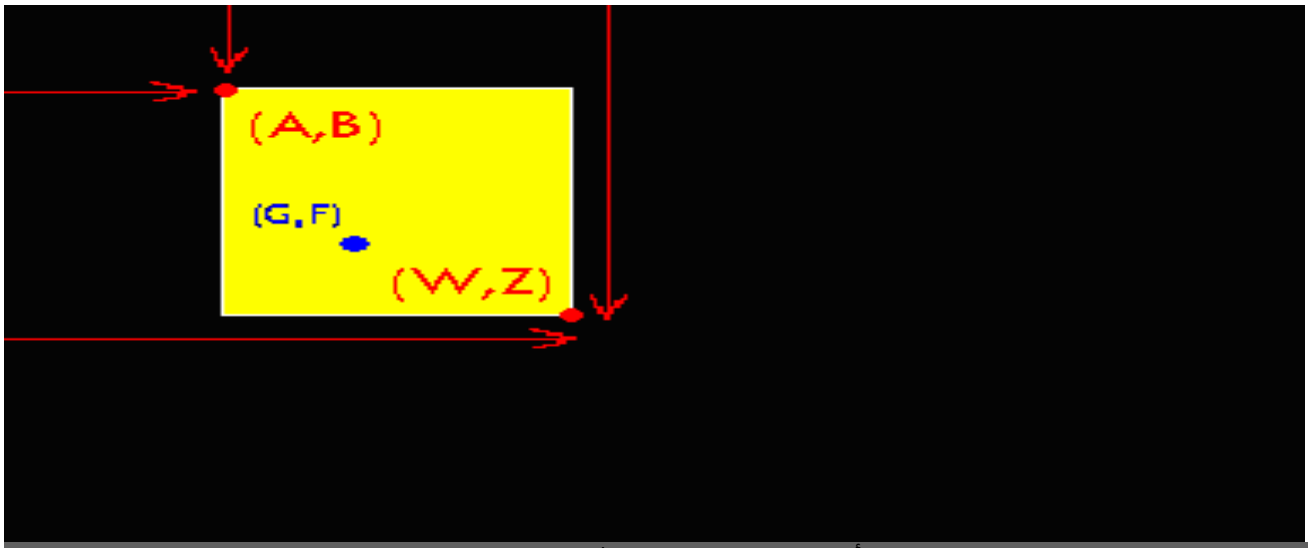
مثال: برنامج لرسم مستطيل وتلوينه.

- rectangle(A,B,W,Z);

إحداثي الركن الأيسر للمستطيل = (A,B)
إحداثي الركن الأيمن للمستطيل = (W,Z)

- floodfill(G,F,getmaxcolor());

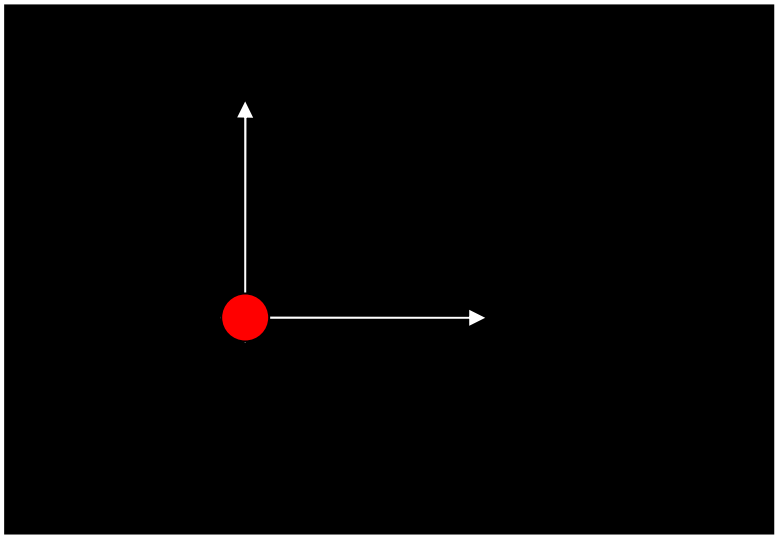
إحداثي نقطة داخل المستطيل = (G,F)



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
  Int a=100,b=200,w=100,z=300,G=150,F=200;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
rectangle(a,b,w,z);
setcolor(14);
floodfill(G,F,getmaxcolor());
getch();
}
```

الفصل الرابع/ النقطة المرجعية



النقطة المرجعية

النقطة المرجعية هي نقطة معلومة ولكن لا تظهر على الشاشة ويمكن معرفة موقعها أثناء البرنامج كما يمكن تغيير مكانها.

دالة لتغيير مكان النقطة المرجعية/ moveto

لتغيير مكان النقطة المرجعية إلى الإحداثي (x,y)

```
moveto(x,y);
```

دالة لإزاحة مكان النقطة المرجعية/ moveto

لإزاحة النقطة المرجعية (مسافة قدرها dx أفقيا ومسافة قدرها dy رأسيا) وذلك نسبة إلى الموقع الأصلي للنقطة المركزية.

```
moverel(dx,dy);
```

دالة للحصول على الإحداثي السيني للنقطة المرجعية/ getx

تمكنك من الحصول على قيمة عددية.
مثلا: لو كان الإحداثي السيني=100 فسوف ترجع الدالة رقم (100) وتخصصه للمتغير:(m مثلا).

```
m=getx();
```

دالة للحصول على الإحداثي الصادي للنقطة المرجعية/ gety

تمكنك من الحصول على قيمة عددية .
مثلا: لو كان الإحداثي الصادي=100 فسوف ترجع الدالة رقم (100) وتخصصه للمتغير:(m مثلا).

```
m=gety();
```

دالة لرسم مستقيم تتابعي/ lineto

لرسم مستقيم من النقطة المرجعية إلى النقطة (x,y) ثم تحويل النقطة الجديدة إلى نقطة مرجعية.

```
lineto(x,y);
```

دالة لرسم مستقيم تتابعي آخر/ linerel

لرسم مستقيم من النقطة المركزية إلى نقطة (تبعد أفقيا مسافة قدرها dx وتبعد مسافة قدرها dy رأسيا) وذلك نسبة إلى النقطة المرجعية، ثم تحويل النقطة الجديدة إلى نقطة مرجعية.

```
linerel(dx,dy);
```

الفصل الخامس/ كتابة النصوص

THIS IS MY TEST

Enter start

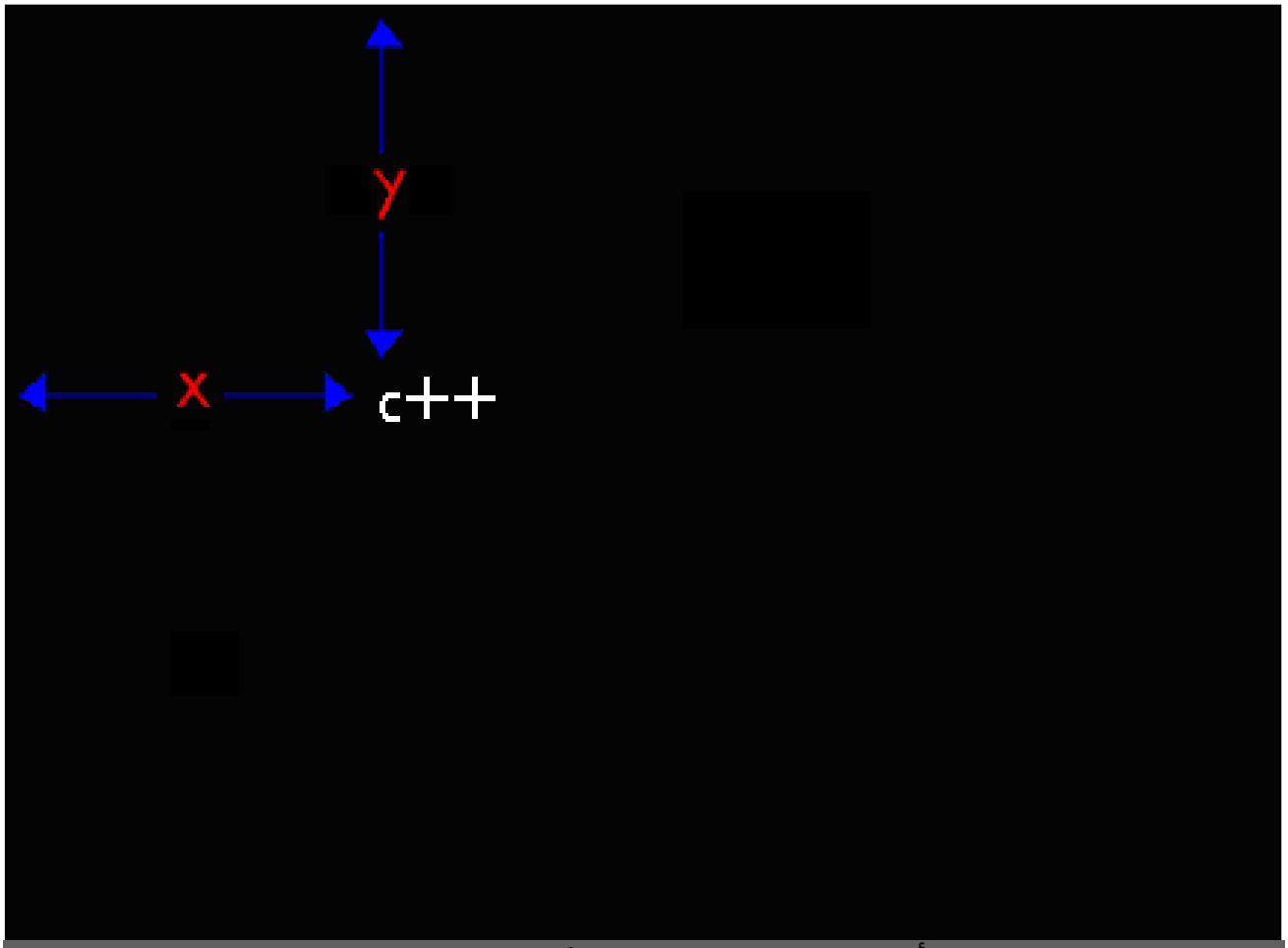
Write your name

دالة لإظهار نص عند النقطة المرجعية/outtext

ملاحظة: ضع النص المراد إظهاره بين علامتي التنصيص في المكان المظلل.
فيظهر النص عند إحداثيات النقطة المركزية (x,y).

ملاحظة: الموقع الإبتدائي للنقطة المركزية هو (0,0) ما لم يتم تغيير مكانها.
ملاحظة: هذه الدالة تطبع النصوص ولا تطبع المتغيرات العددية.

outtextxy("c++");



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

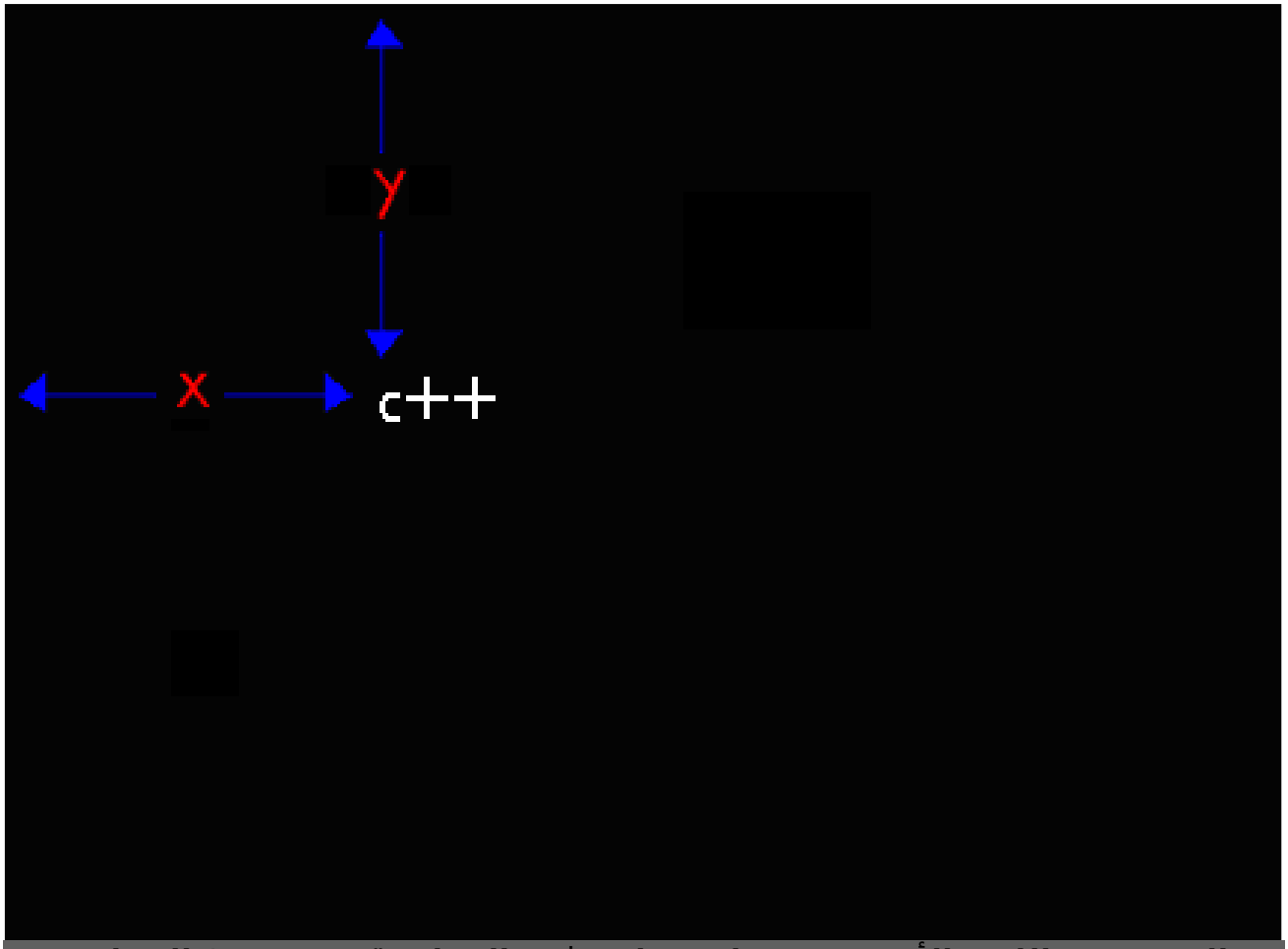
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int x=45,y=60,h=15;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
outtextxy("c++");
getch();
}
```

دالة إظهار نص عند نقطة معينة/ outtextxy

تحتوي على 2 متغيرات عددية هما: إحداثيات النقطة (x,y).

ملاحظة: ضع النص المراد إظهاره بين علامتي التنصيص في المكان المظلل.
ملاحظة: هذه الدالة تطبع النصوص ولا تطبع المتغيرات العددية.

outtextxy(x,y,"c++");



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int x=45,y=60,h=15;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
outtextxy(x,y,"c++");
getch();
}
```

دالة تغيير حجم ونوع النص/ settextstyle

تحتوي على 3 متغيرات عددية:
A قيمة عددية للحجم (الأحجام مرتبة تصاعديا من 1 إلى 11)
B قيمة عددية لاتجاه النص (0 للاتجاه الأفقي بينما 1 للاتجاه العمودي)،
ملاحظة: "النص باللغة الإنجليزية"
C قيمة عددية لنوع الخط (نوع الخط مرتب تصاعديا من 1 إلى 10 تقريبا)

```
settextstyle(A,B,C);
```

تنبيه: تكتب (هذه الدالة) قبل دالة (كتابة النص).

دالة لتخزين القيم العددية في مصفوفة نصية/ sprintf

تستخدم هذه الدالة لتحويل الأعداد إلى مصفوفة نصية، حتى نستطيع طباعة الأعداد باستخدام دالة outtext

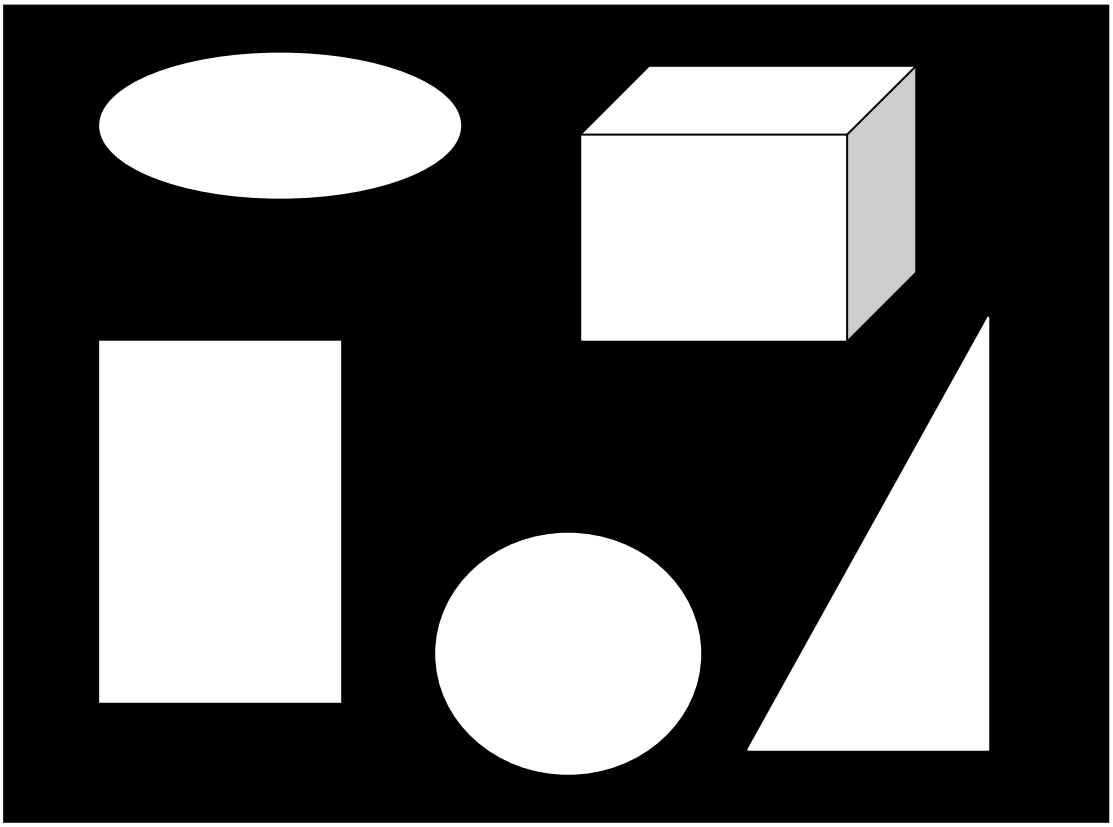
```
sprintf(msg,"%d %d",a,b);
```

a متغير صحيح.
b متغير صحيح.
msg مصفوفة من نوع char

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
char msg[10];
int a=12,b=10;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
sprintf(msg,"%d %d",a,b);
outtextxy(x,y,msg);
getch();
}
```

الفصل السادس/

الأشكال الهندسية المسطحة

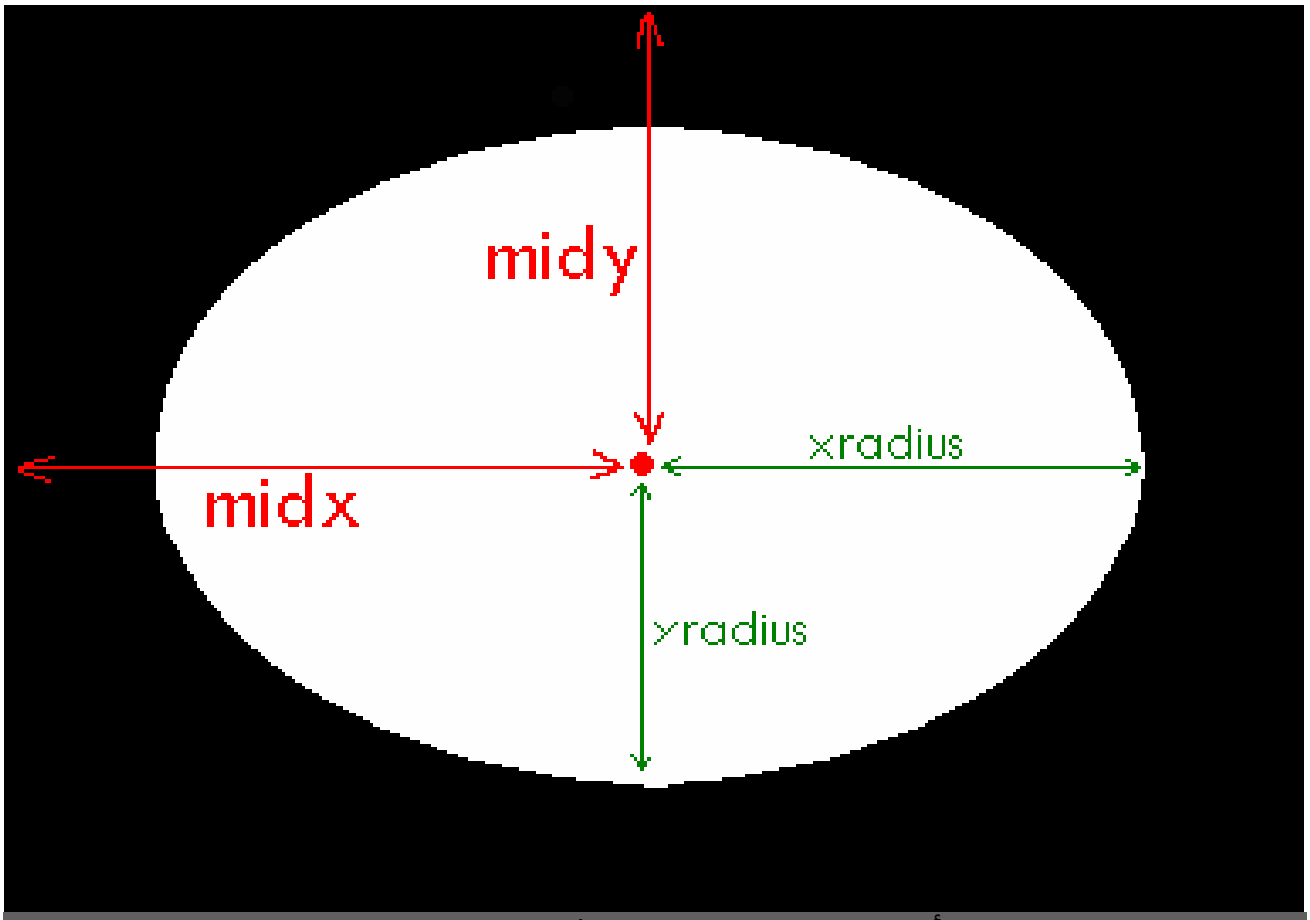


دالة رسم قطع ناقص مصمت/ fillellipse

تستخدم لرسم قطع ناقص مع طلائه من الداخل باللون المطلوب.
احداثي المركز=(midx, midy)
نصف قطر الإحداثي x =Xradius
نصف قطر الإحداثي y =Yradius

ملاحظة: تشبه دالة رسم القطع الناقص فيما عدا أنها لا تتضمن زاويتي البدء والنهاية، لأنها ترسم قطع ناقص.

Fillellipse(midx, midy, xradius, yradius);



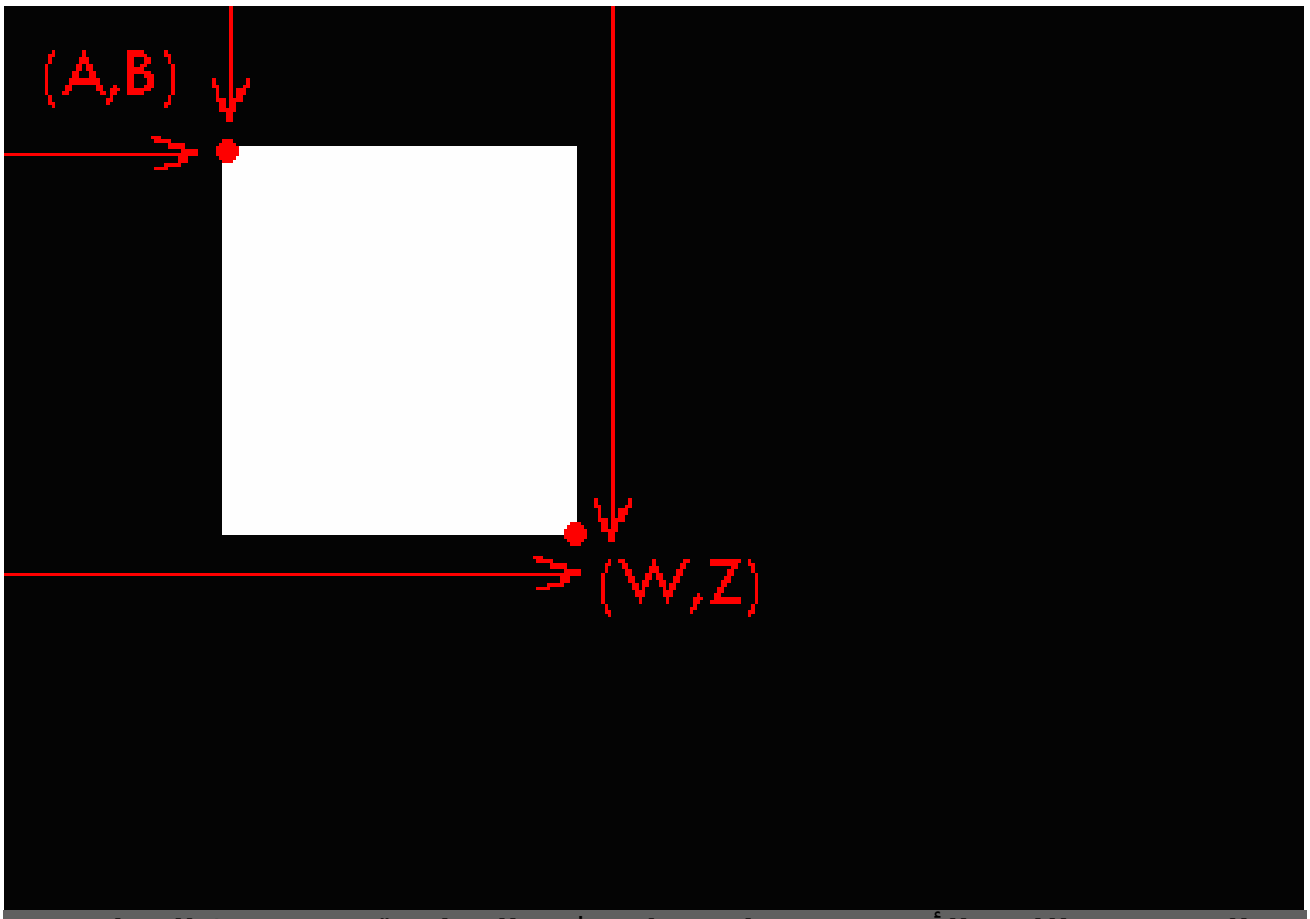
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
   int midx=320, midy=240, stangle = 45;
   int endangle = 135, radius = 100;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
fillellipse (midx, midy, xradius,yradius);
getch();
}
```

دالة رسم مستطيل مصمت/ bar

إحداثي أقصى اليسار= (A,B)
إحداثي أقصى اليمين= (W,Z)

bar(A,B,W,Z);



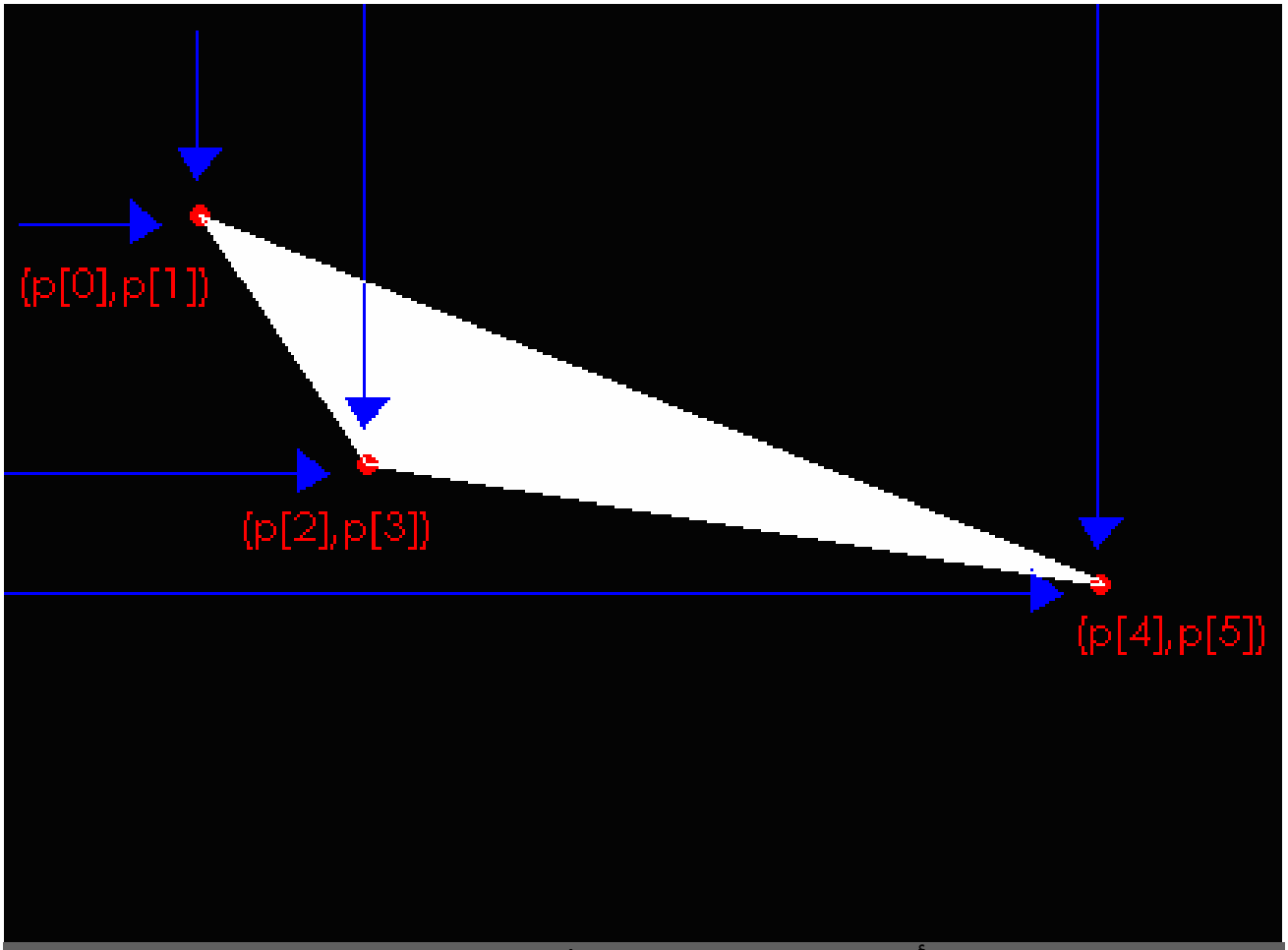
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
  Int a=10,b=20,w=150,z=200;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
bar(a,b,w,z);
getch();
}
```

دالة رسم أشكال مضلعة مصمته/ fillpoly

تستخدم لرسم شكل مضلع مصمت بالإضافة لتلوينه.
اسم مصفوفة النقاط هي p[n] حيث n عدد النقاط.

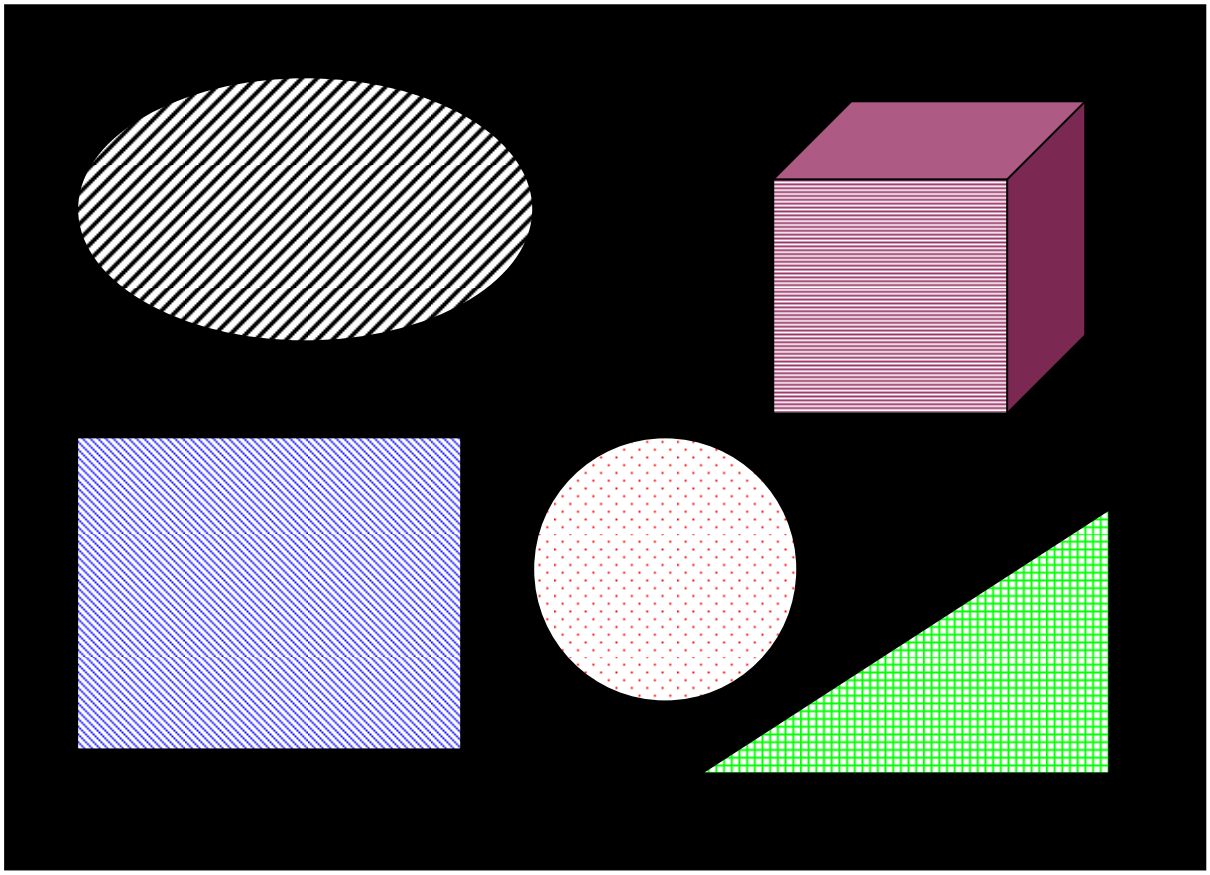
Fillpoly(n,p);



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
Int p[3];
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
P[0]=10;p[1]=15;
P[2]=100;p[3]=200;
P[4]=200;p[5]=250;
fillpoly(3,p);
getch();
}
```

الفصل السابع/ تلوين الأشكال المسطحة

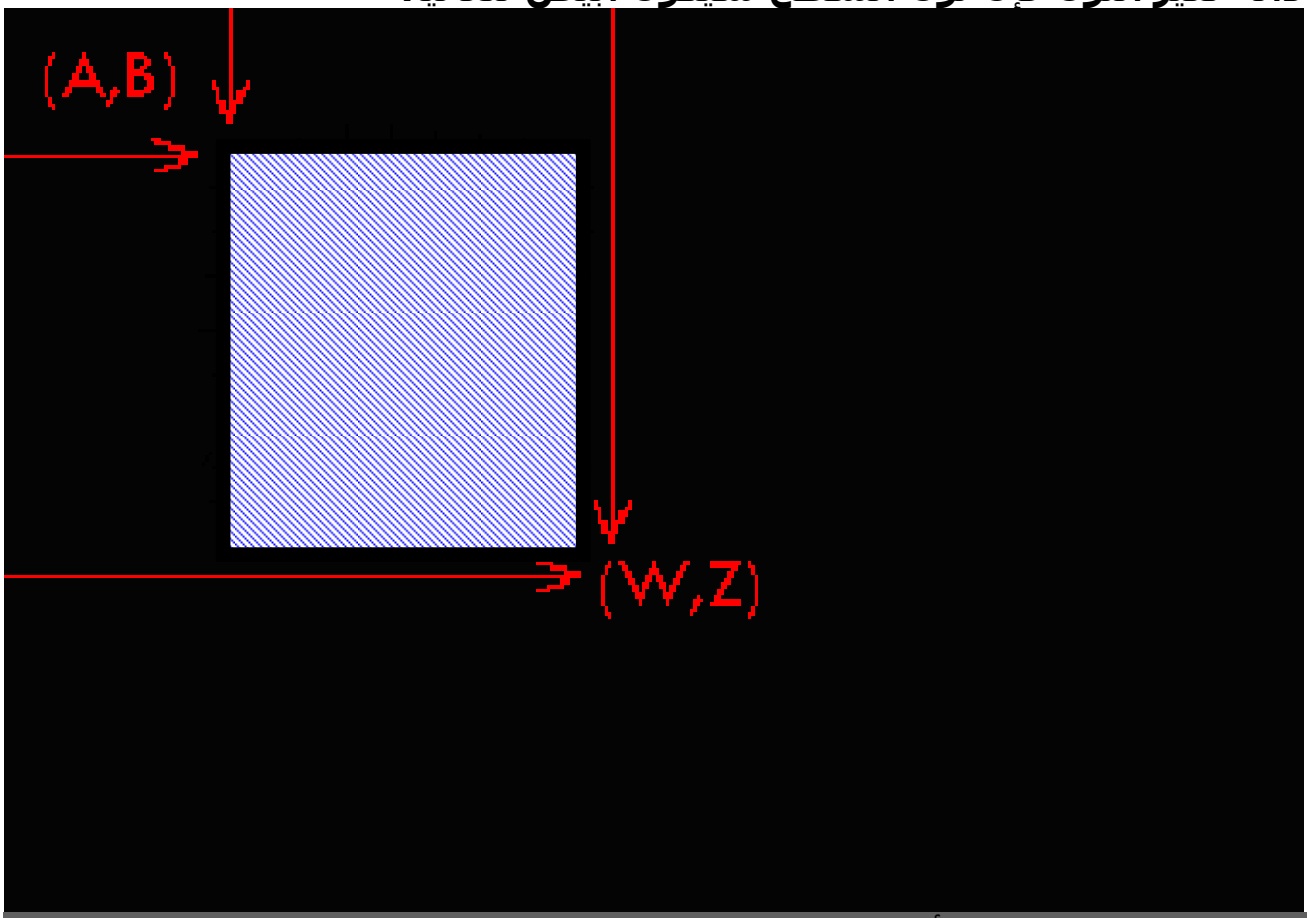


دالة تغيير نوع ولون السطح المصمت/ setfillstyle

تستخدم لملء المساحات باللون المطلوب.
يمكنك تغيير لون ونوع السطح للأشكال المرسومة باستخدام هذه الدالة:
K عدد صحيح من 0 إلى 12 يرمز لنوع السطح
C عدد صحيح من 0 إلى 15 يرمز للون السطح

setfillstyle(k,c);

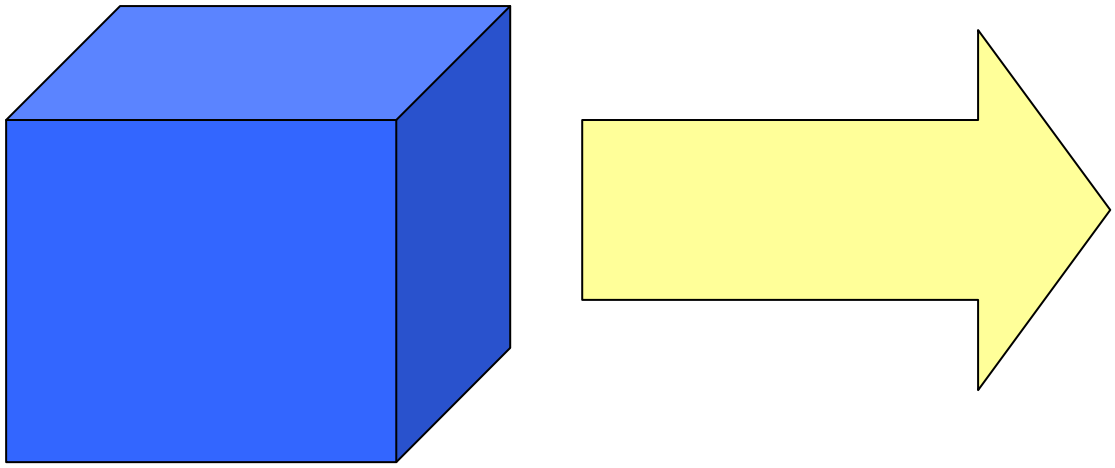
ملاحظة: يجب أن تكتب (هذه الدالة) قبل (دالة الرسم)، وإذا لم تستعمل دالة تغير اللون فإن لون السطح سيكون أبيض تلقائيا.



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
  Int a=10,b=20,w=150,z=200;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
setfillstyle(1,15);
bar(a,b,w,z);
getch();
}
```

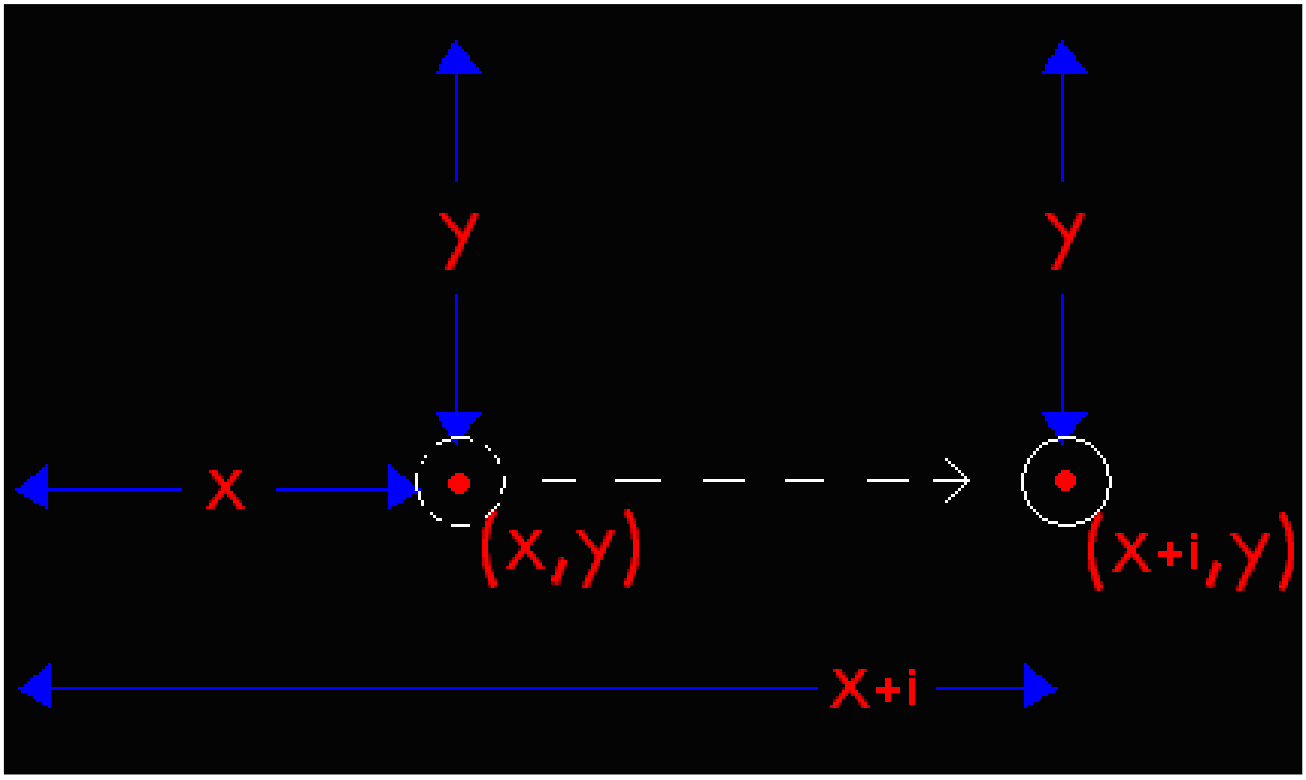
الفصل الثامن/ طرق الإزاحة



تصميم برنامج لتحريك دائرة أفقيا

- إذا أردنا تحريك نقطة أفقيا جهة اليمين من (x1,y) إلى (x2,y).

1. فإننا نرسم الدائرة في (x1,y)، ونصف قطرها h
2. ثم نثبتها على الشاشة لحظة، وذلك باستخدام دالة ;(100)delay وهي تقوم بتثبيت الشاشة لمدة 10ms حيث (الثانية الواحدة= 1000ms)، ويكتب مقدار الزمن بين قوسي الدالة في المكان المظلل.
3. ثم نقوم بمسحها وذلك بأن نرسم نفس الدائرة و لكن باللون الأسود (0)
4. ثم نقوم برسم الدائرة مرة أخرى ولكن بإزاحة قدرها ;x=x+1
5. ونكرر هذه العمليات عدة مرات حتى نصل لمقدار الإزاحة المطلوبة.



#include<stdio.h> #include<conio.h> #include<dos.h> #include<graphics.h>	مكتبة الدالة ;(100)delay

void main()
{
int x=45,y=60,h=3;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");	

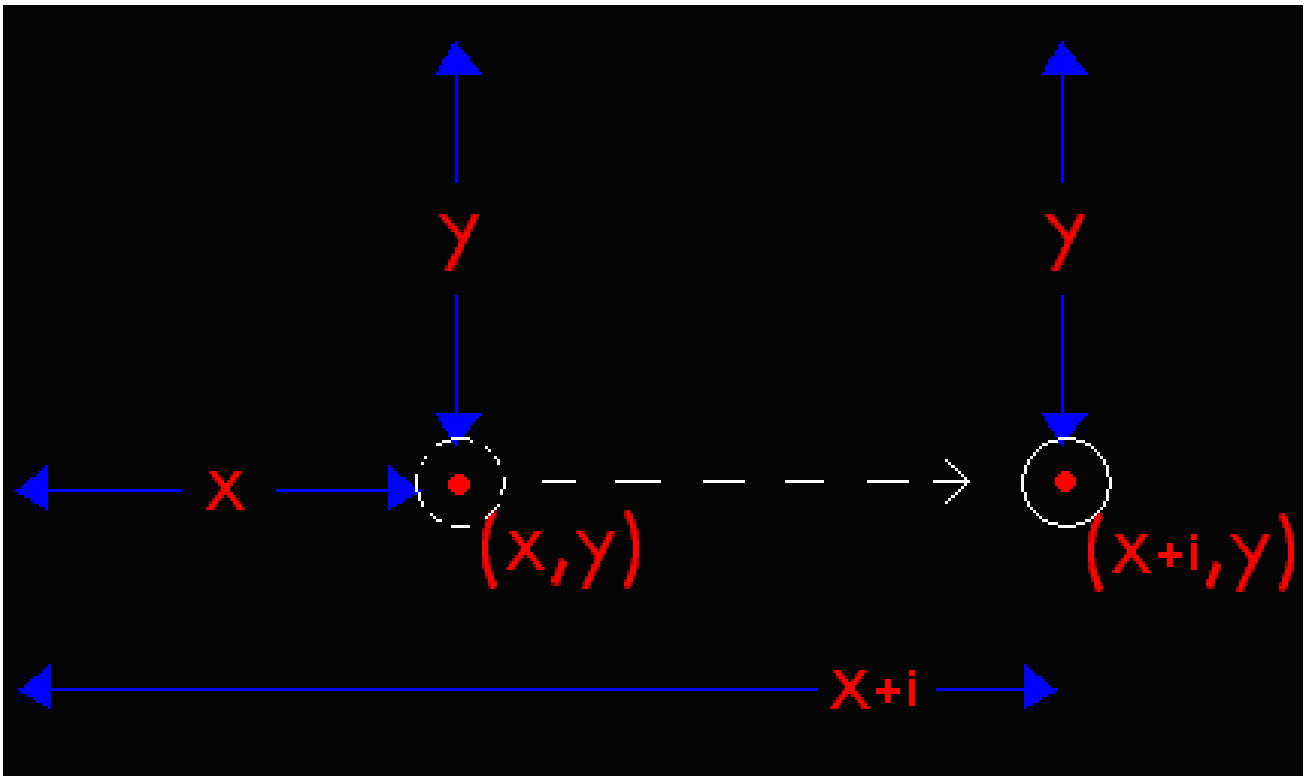
for(int i=0;i<100;i++)
{
Setcolor(0); circle(x,y,h);
X=x+1
Setcolor(15); circle(x,y,h);
delay(100);
}
	مقدار الإزاحة من 1 إلى 100 نقطة.
	
	لمسح الدائرة (نرسمها باللون الأسود)
	إزاحة أفقية 1
	لرسم الدائرة (نرسمها باللون الأبيض)
	دالة تثبيت الشاشة لمدة 100 مللي ثانية

getch(); }	

طريقة أخرى لتحريك دائرة أفقيا

- إذا أردنا تحريك نقطة أفقيا جهة اليمين من (x1,y) إلى (x2,y).

1. نرسم الدائرة وليكن مركزها هو (x1,y) ونصف قطرها h
2. ثم نثبتها على الشاشة لحظة، وذلك باستخدام دالة ;(100)delay وهي تقوم بتثبيت الشاشة لمدة 10ms حيث (الثانية الواحدة= 1000ms)، ويكتب مقدار الزمن بين قوسي الدالة في المكان المظلل.
3. ثم نقوم بمسح الشاشة كلها باستخدام دالة مسح الشاشة/ ;() cleardevice
4. ثم نقوم برسم الدائرة مرة أخرى ولكن بإزاحة قدرها x=x+1
5. ونكرر هذه العمليات عدة مرات حتى نصل لمقدار الإزاحة المطلوبة.



#include<stdio.h> #include<conio.h> #include<dos.h> #include<graphics.h>	مكتبة الدالة ;(100)delay

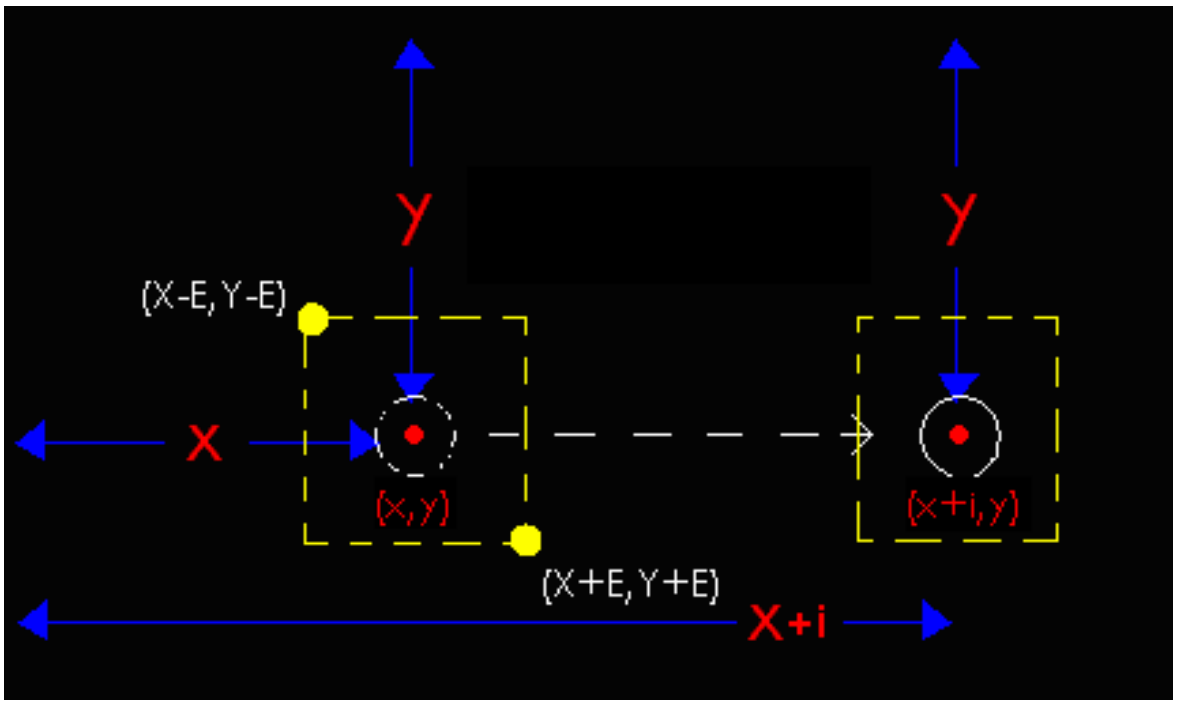
void main()
{
int x=45,y=60,h=3;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");	

for(int i=0;i<100;i++)
{
cleardevice ();
x+=1;
circle(x,y,h);
delay(100);
}
	مقدار الإزاحة من 0 إلى 100 نقطة.
	
	دالة مسح الشاشة
	إزاحة أفقية 1
	
	دالة تثبيت الشاشة لمدة 100 مللي ثانية

getch(); }	

تصميم برنامج لتحريك دائرة أفقيا (وإيقاف الحركة عند أي لحظة)

- يقوم البرنامج بعمل نسخة من الشكل في الذاكرة ثم يقوم بمسح الشكل من الشاشة ثم يرسم الشكل مرة أخرى ولكن بإزاحة قدرها x=x+1;، ويتوقف البرنامج عند الضغط على أي زر




```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>
void draw_arrow(int x, int y);
int main(void)
{
int gdriver = DETECT, gmode, errorcode;
void *arrow;
int x, y, E=10, i=0;
unsigned int size;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
x = 45;
y = 60;
draw_arrow(x, y);
size = imagesize(x-E, y- E, x+ E, y+ E);
arrow = malloc(size);
getimage(x-E, y- E, x+ E, y+ E, arrow);
while (!kbhit())
{
putimage(x-E, y- E, arrow, XOR_PUT);
x += 1;
if (i>100) x = 45;
putimage(x-E, y- E, arrow, XOR_PUT);
delay(100);
i+=1;
}
free(arrow);
closegraph();
return 0;
}
void draw_arrow(int x, int y)
{
int h=3;
circle(x,y,h);
}
```