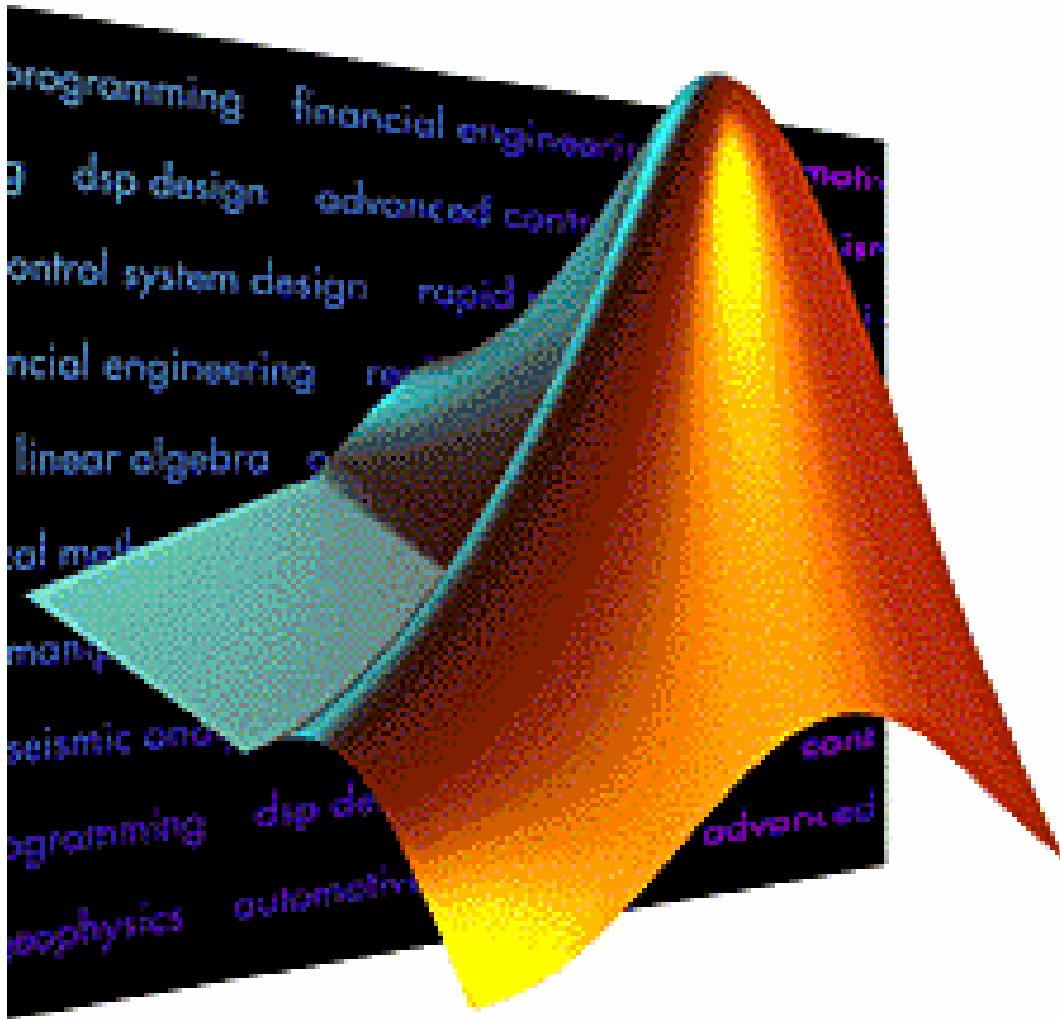


# برمجة الرسم بلغة سي

باستخدام **TURBO C++**



تأليف: البراء عبد الرؤوف الرملي

[albararamli@yahoo.com](mailto:albararamli@yahoo.com)

# المحتويات

الصفحة	شرح
4	مفاهيم أساسية
7	الفصل الأول/ دوال الرسم الموجودة في <code>#include&lt;graphics.h&gt;</code>
11	الفصل الثاني / الأشكال الهندسية
19	الفصل الثالث/ تلوين الأشكال الهندسية
22	الفصل الرابع / النقطة المرجعية
24	الفصل الخامس/ كتابة النصوص
28	الفصل السادس/ الأشكال الهندسية المسطحة
32	الفصل السابع/ تلوين الأشكال الهندسية المسطحة
34	الفصل الثامن/ طرق الإزاحة

## حديث نبوي

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

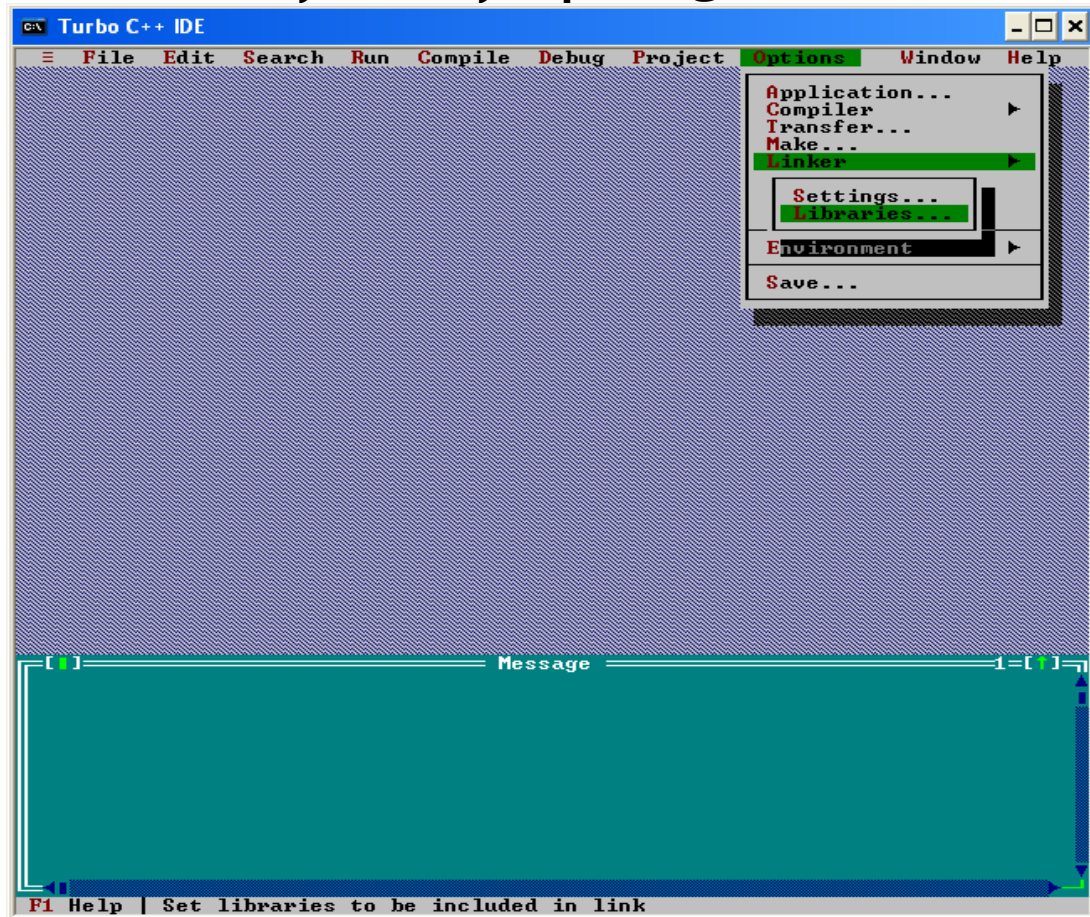
( كل مصور في النار يجعل له بكل صورة صورها نفسا فيعذبه في جهنم ) . - متفق عليه -

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

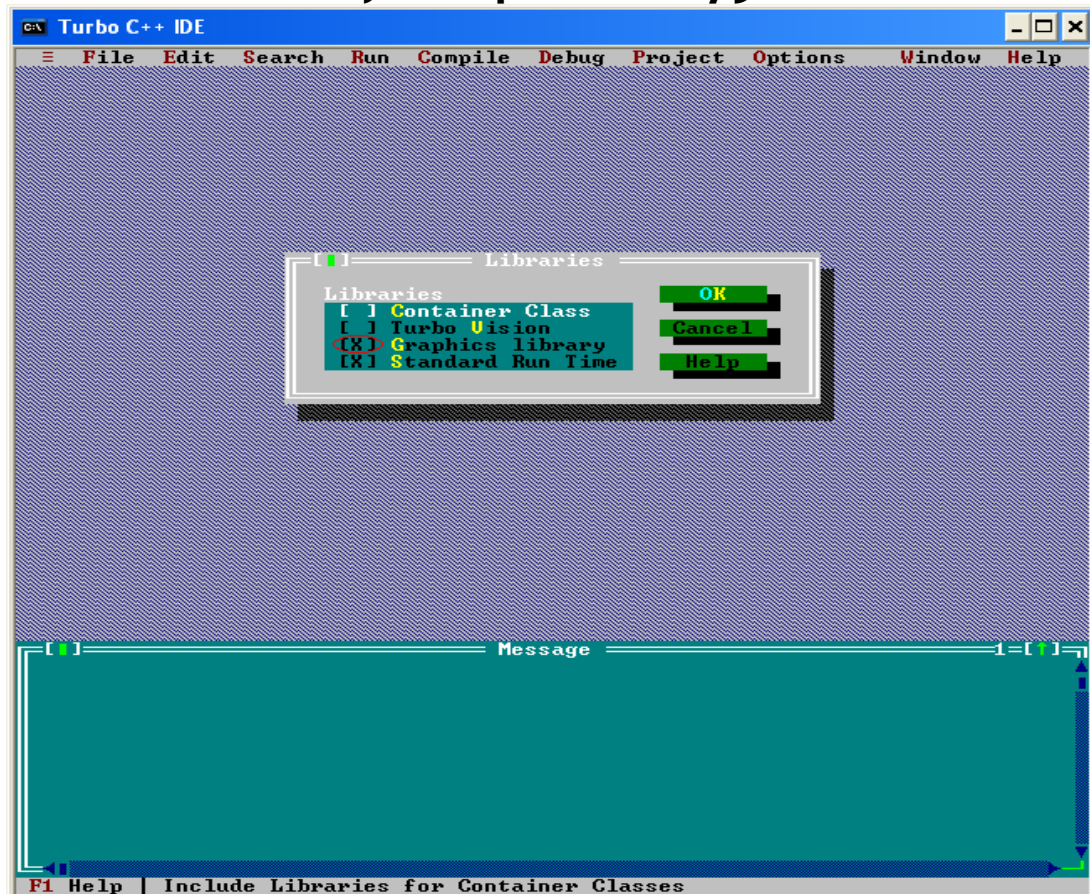
# مفاهيم أساسية



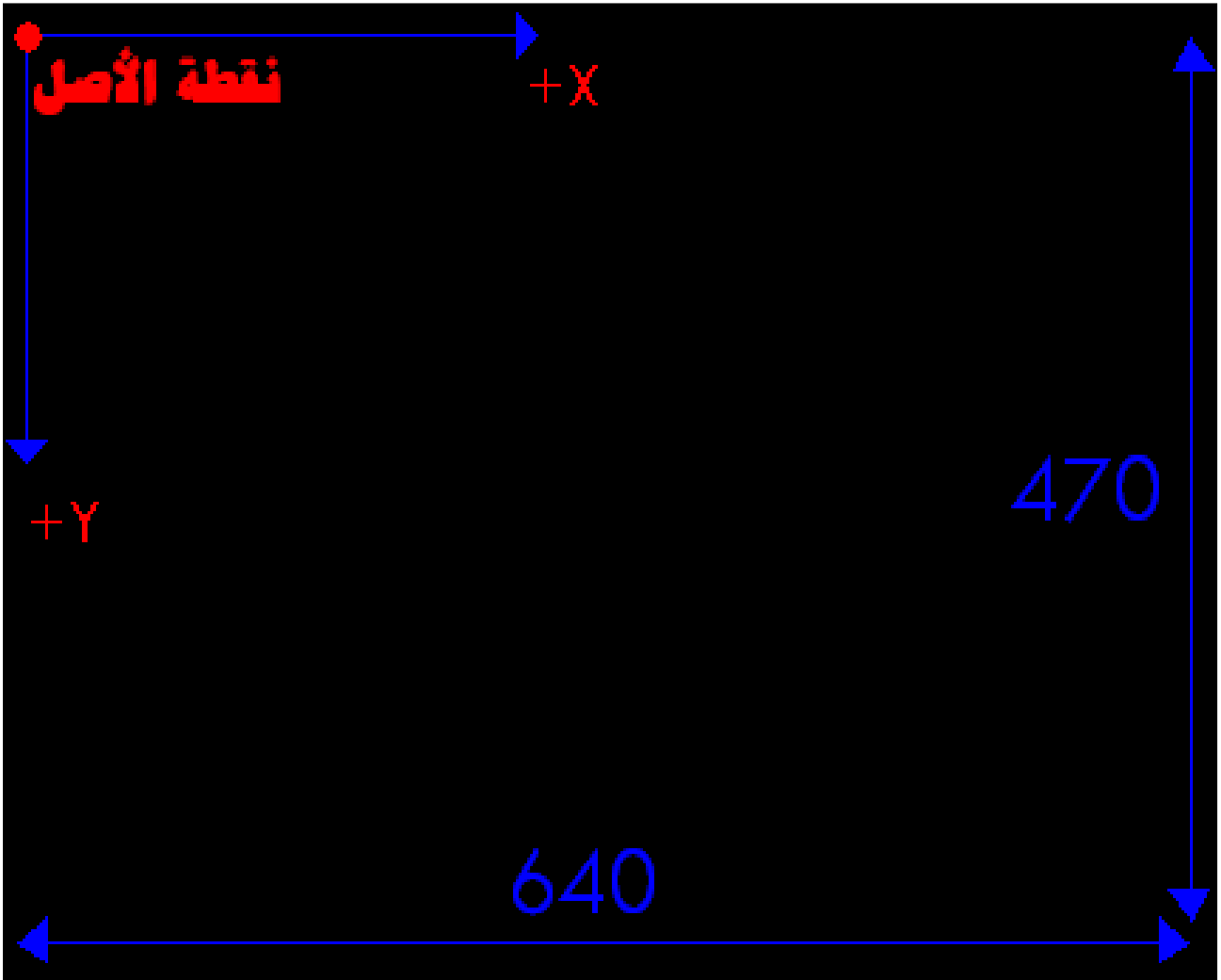
## ▪ اضغط على Options ثم Linker ثم Libraries



## ▪ 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 =  $640 \times 480$
2. وال (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:

<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt;</pre>	
<pre>#include&lt;graphics.h&gt;</pre>	استدعاء مكتبة الرسم
<pre>void main() {</pre>	
<pre>int gdriver = DETECT, gmode, errorcode;</pre>	تؤدي للتعرف على كارت الرسم الموجود
<pre>initgraph(&amp;gdriver, &amp;gmode, "c:\\tc\\bgi");</pre>	<p>بمجرد استدعاء هذه الدالة يصبح الجهاز قادر على استقبال الأوامر من دوال الرسم المختلفة. تنبيه: يجب كتابة اسم الممر <code>c:\\tc\\bgi</code> على حسب مكان TURBO C++ في القرص الصلب.</p>
<pre>..... ..... ..... (دوال الرسم تكتب هنا) ..... ..... .....</pre>	
<pre>getch(); }</pre>	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 مظلة باللون البنفسجي وهي 35 دالة

arc	imagesize
bar	initgraph
bar3d	installuserdriver
circle	installuserfont
cleardevice	line
clearviewport	linereel
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 `initgraph(&gdriver, &gmode, "c:\\tc\\bgi")` التي تؤدي لفتح نافذة الرسم.

### دالة مسح الشاشة / `cleardevice`

يمكنك مسح الشاشة بإدراج هذه الدالة:

**`cleardevice ();`**

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

### دالة للحصول على أقصى إحداثي سيني للشاشة / `getmaxx`

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

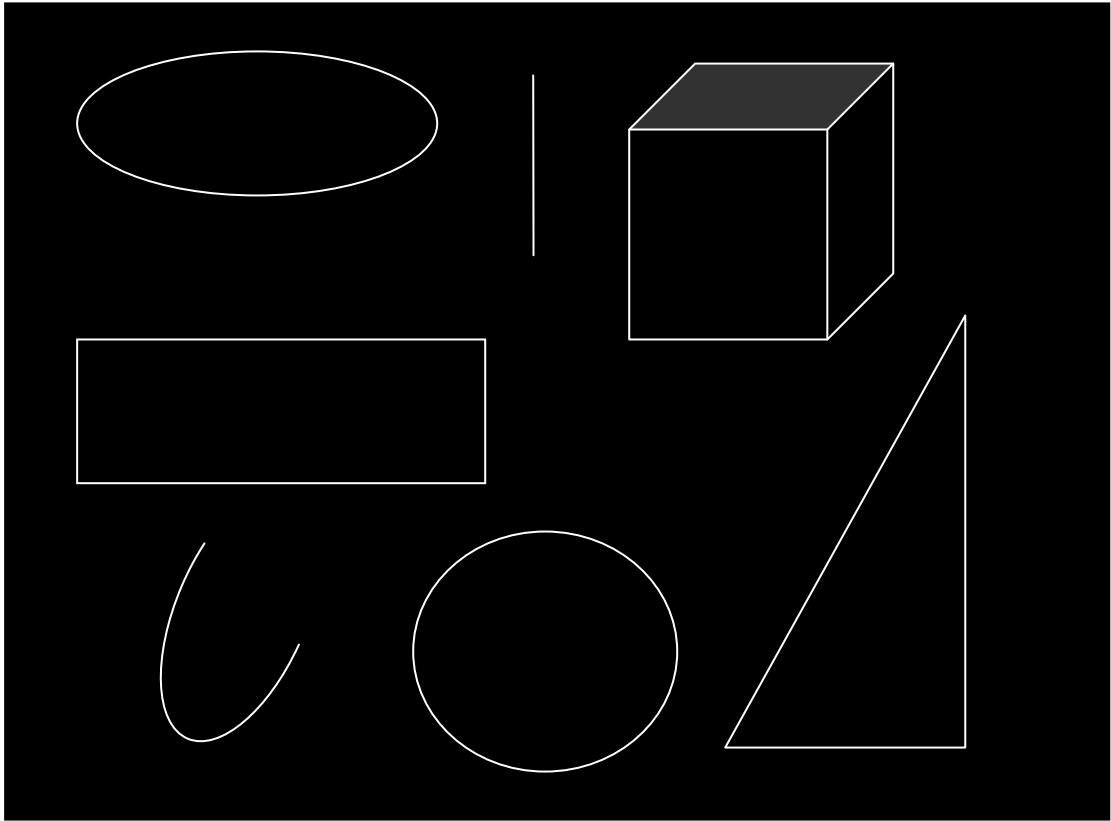
**`m=getmaxx();`**

### دالة للحصول على أقصى إحداثي صادي للشاشة / `getmaxy`

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

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

# الفصل الثاني / الأشكال الهندسية

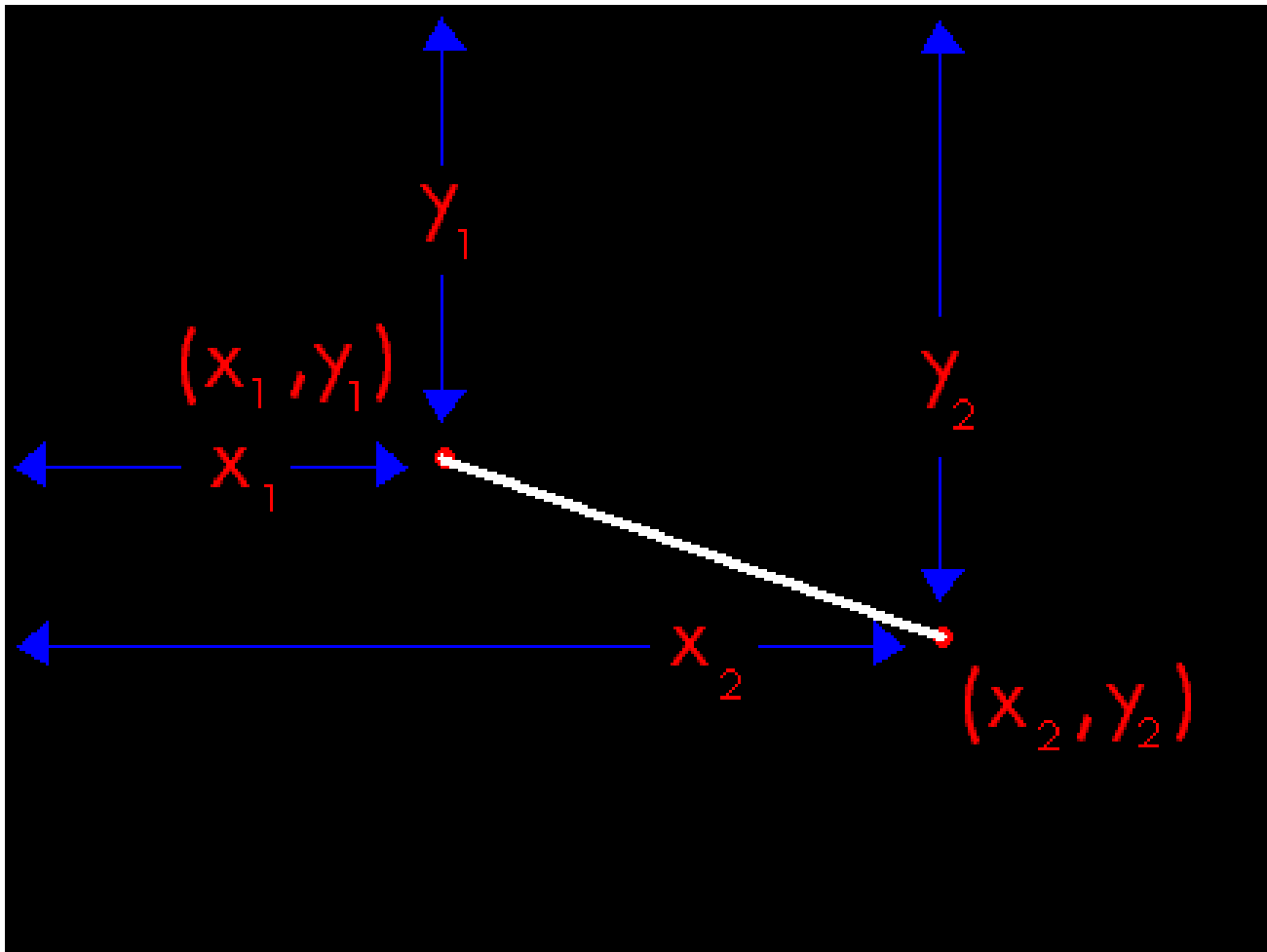


## دالة رسم خط مستقيم / line

تحتوي على 4 متغيرات عددية هم: إحداثي النقطة الأولى  $(x_1, y_1)$  وإحداثي النقطة الأخيرة  $(x_2, y_2)$ .

ملاحظة: يمكن أن تكون الإحداثيات متغيرات صحيحة أو كسور عشرية.

**Line(x1,y1,x2,y2);**



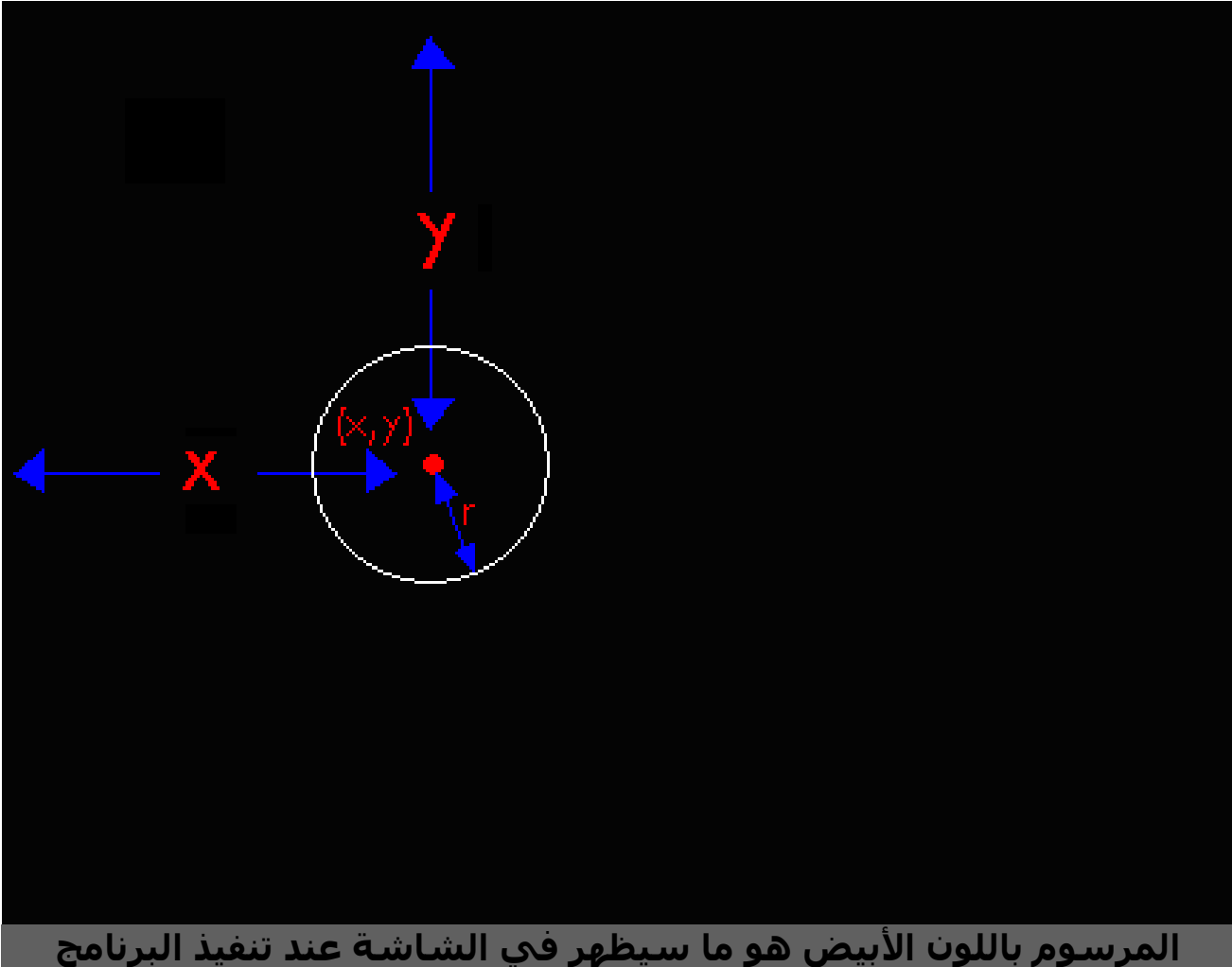
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

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

## 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);**



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

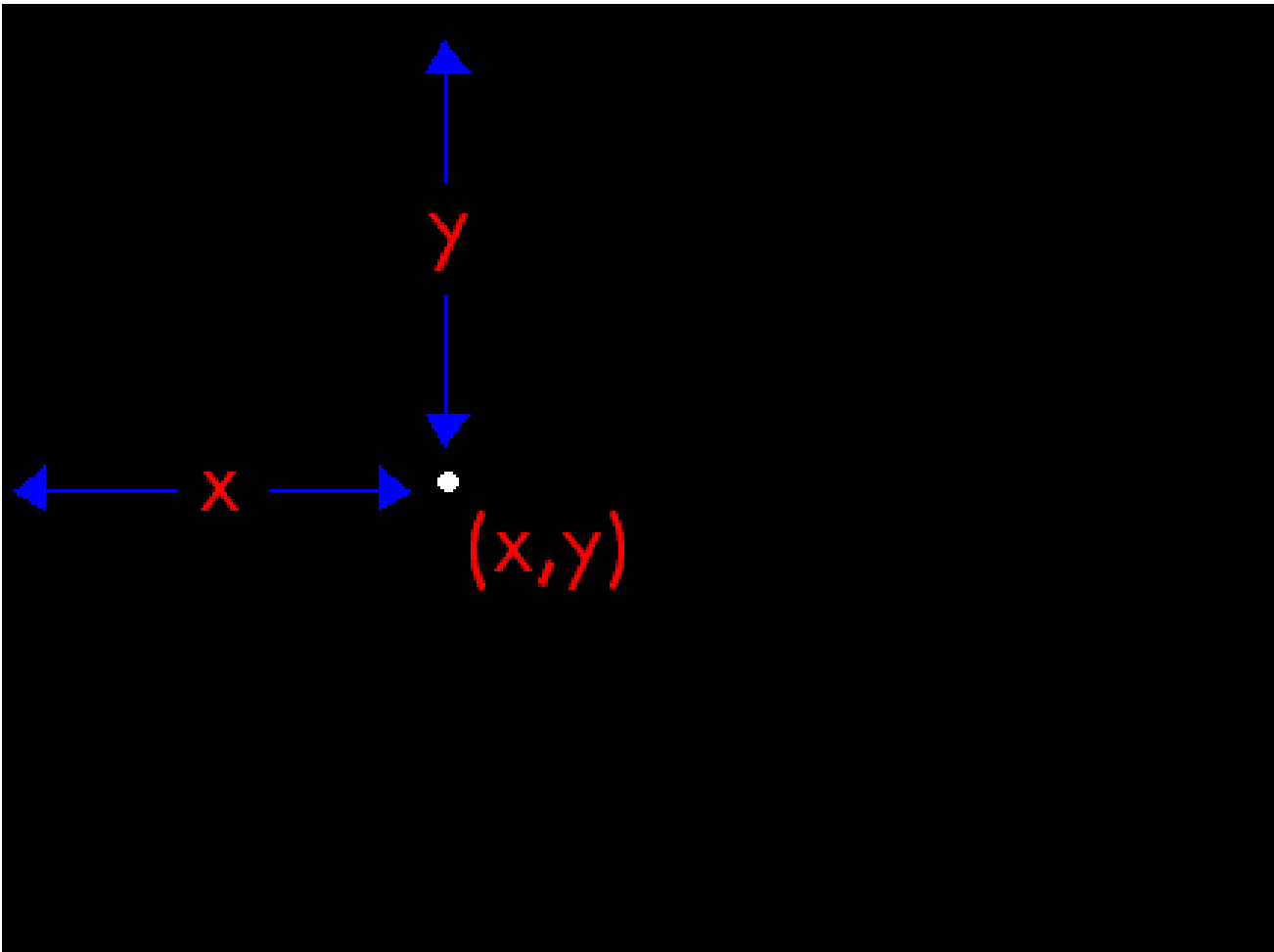
## دالة رسم نقطة / putpixel

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);**



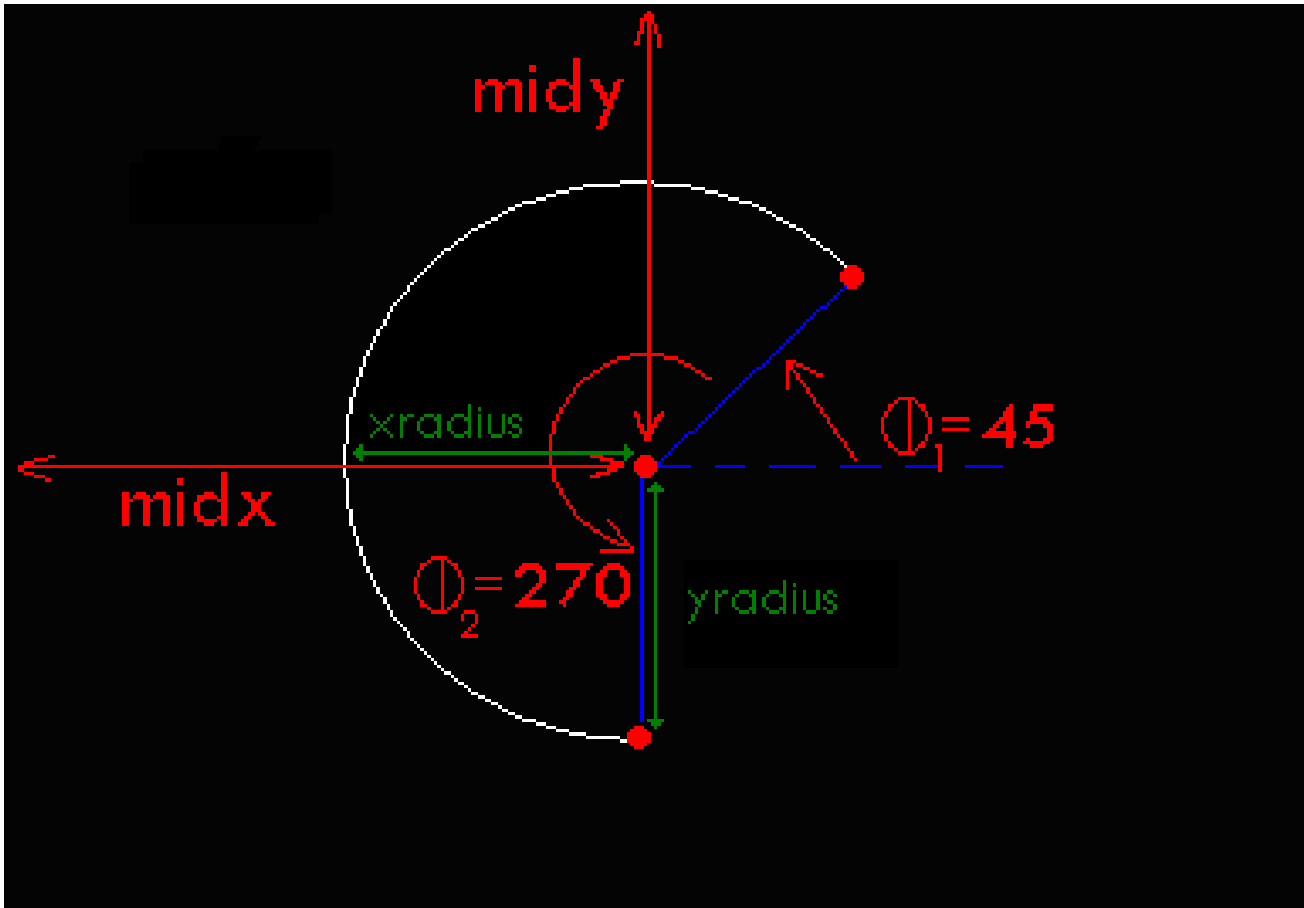
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

#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");	
<b>putpixel(x,y,h);</b>	
getch();	
}	

## دالة رسم أقواس دائرية / arc

(midx, midy) = إحداثي المركز  
 Stangle = ( $\Phi_1$ ) زاوية البدء  
 Endangle = ( $\Phi_2$ ) زاوية النهاية  
 Radius = نصف القطر

**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");	
<b>arc(midx, midy, stangle, endangle, radius);</b>	
getch();	
}	

## دالة رسم قطع ناقص / ellipse

إحداثي المركز = (midx, midy)

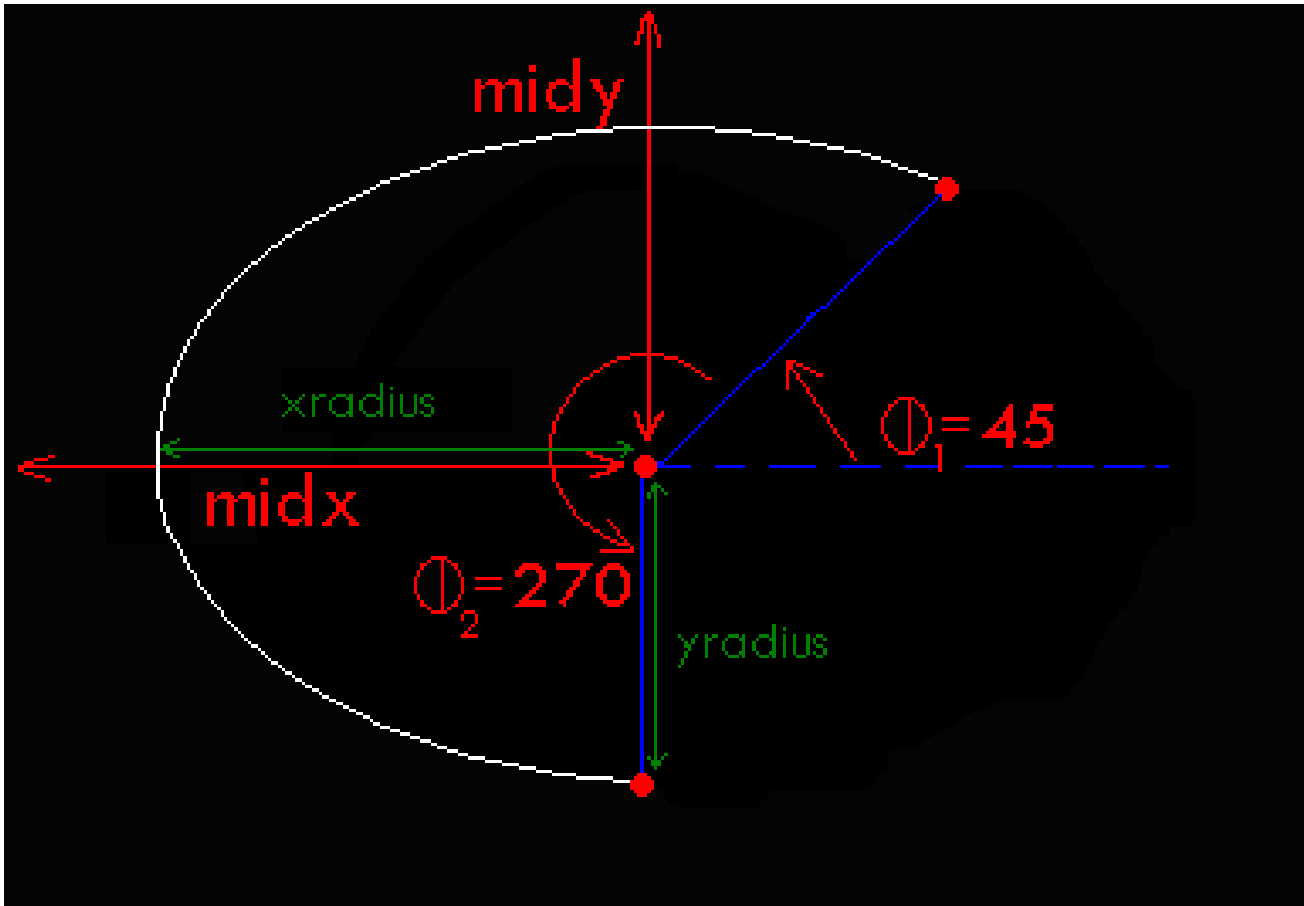
زاوية البدء = Stangle = ( $\Phi_1$ )

زاوية النهاية = Endangle = ( $\Phi_2$ )

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

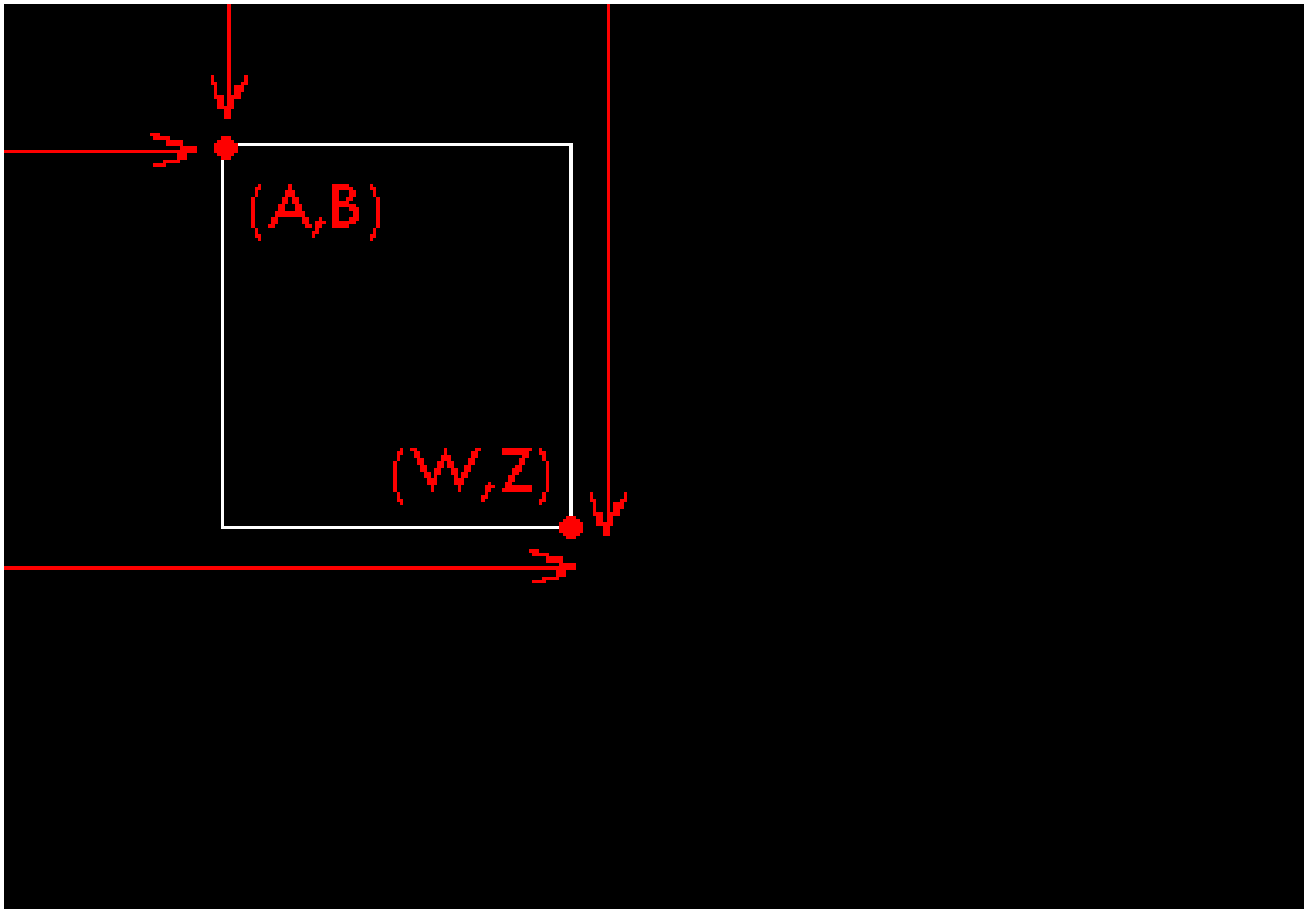
<code>#include&lt;stdio.h&gt;</code>	
<code>#include&lt;conio.h&gt;</code>	
<code>#include&lt;graphics.h&gt;</code>	
<code>void main()</code>	
<code>{</code>	
<code>int midx=320, midy=240, stangle = 45;</code>	
<code>int endangle = 135, radius = 100;</code>	
<code>int gdriver = DETECT, gmode, errorcode;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



## دالة رسم مستطيل / rectangle

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

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



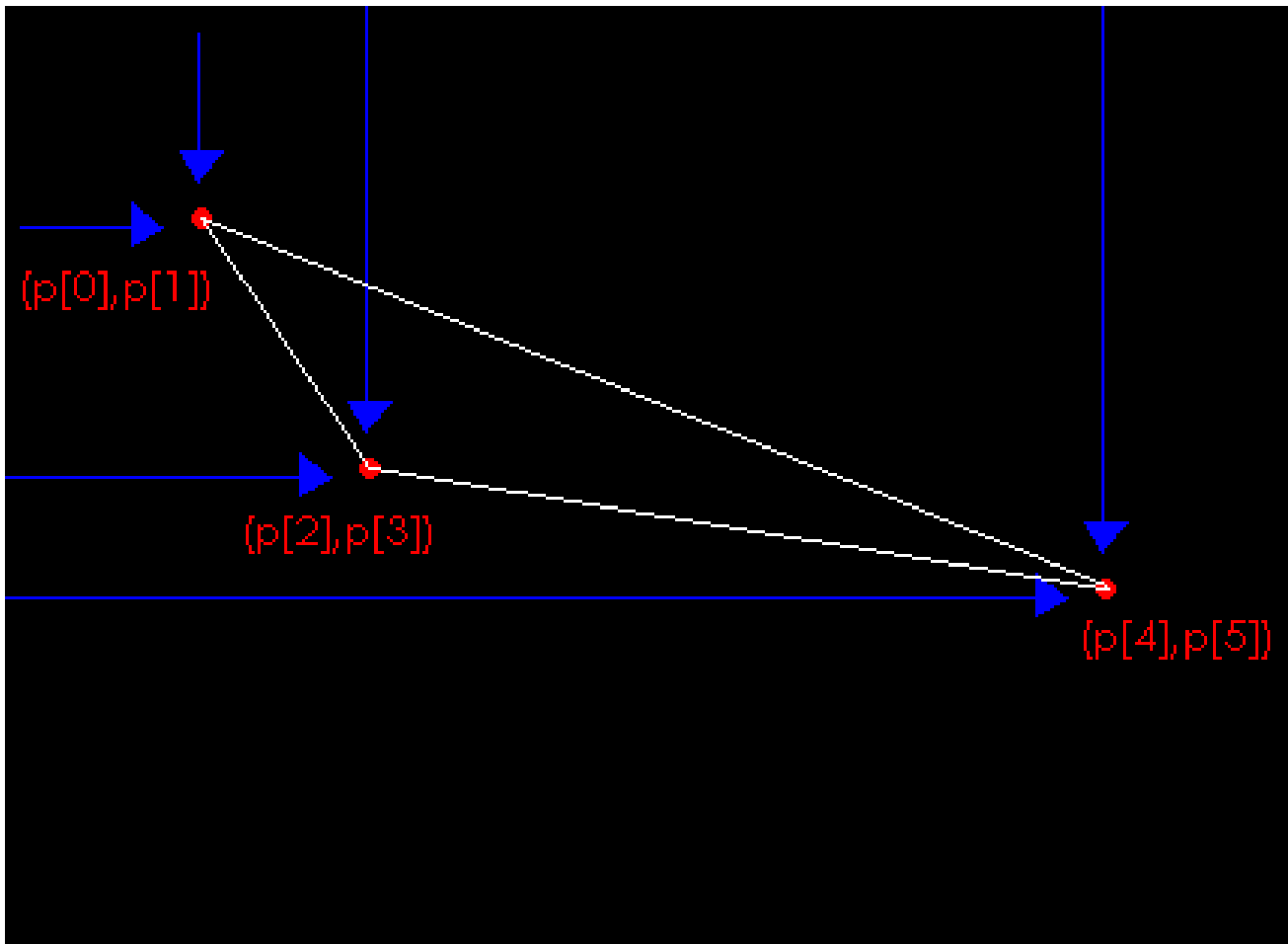
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

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

## دالة رسم الأشكال المضلعة / drawpoly

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

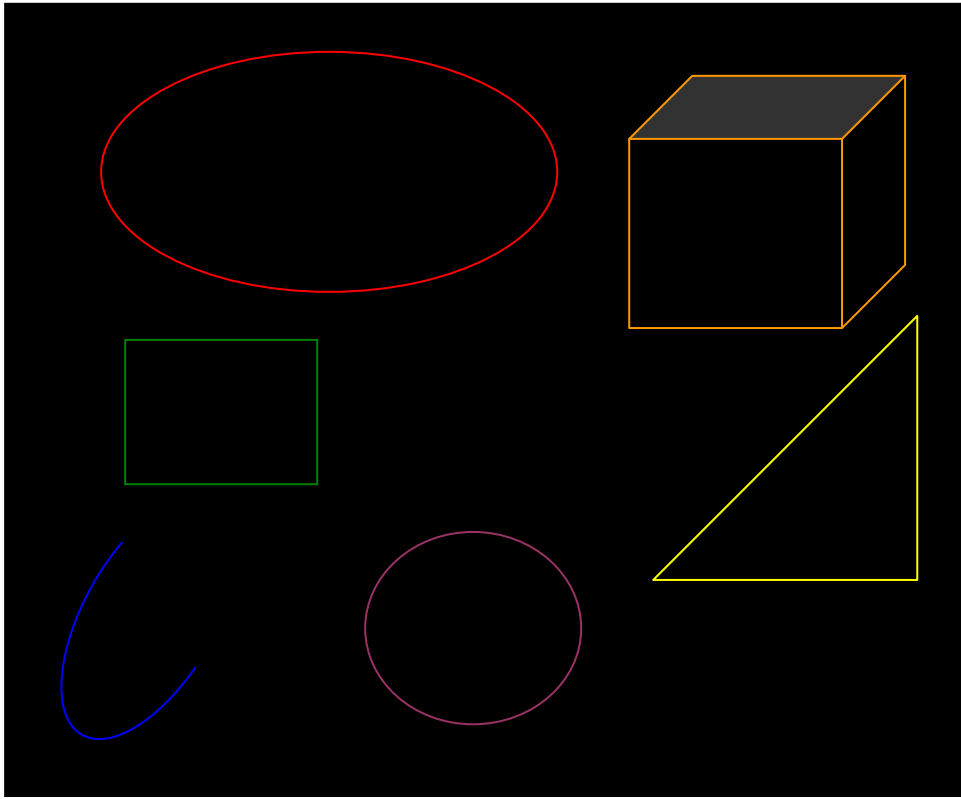
**Drawpoly(n,p);**



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

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

# 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 تلقائياً.

## دالة تغير لون الرسم / Setcolor

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

**setcolor( 15 );**

5	4	3	2	1	0
بنفسجي	أحمر	كحلي	أخضر	أزرق	أسود
11	10	9	8	7	6
كحلي فاتح	أخضر فاتح	أزرق فاتح	رصاصي غامق	رصاصي فاتح	بني
15	14	13	12		
أبيض	أصفر	بنفسجي فاتح	أحمر فاتح		

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

## دالة تغير لون خلفية الشاشة / setbkcolor

يمكنك تحديد لون الخلفية باستخدام الدالة setbkcolor وذلك بإدراج رقم اللون بين قوسى 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 تلقائياً.

## دالة للحصول على لون الخلفية / getbkcolor

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

**m=getbkcolor();**

## دالة للحصول على لون خط الرسم / getcolor

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

**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) وتخصه 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());`

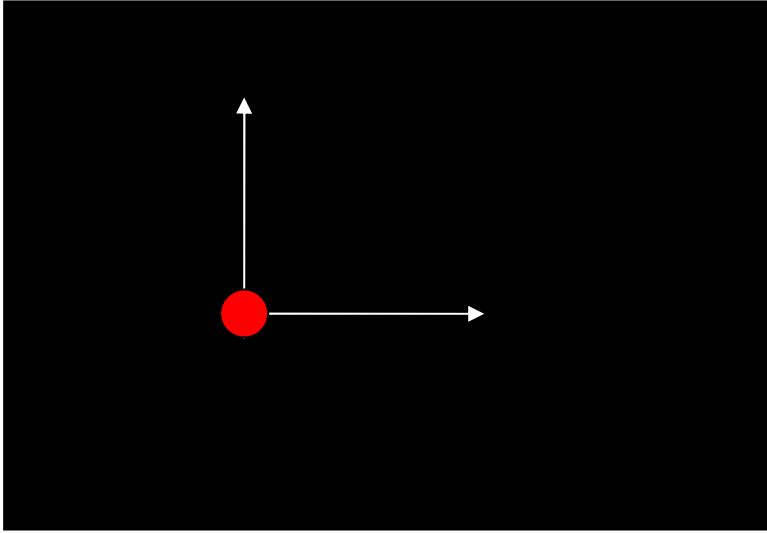
إحداثي نقطة داخل المستطيل =  $(G,F)$



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

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

# 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 / moverel

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

**moverel(dx,dy);**

### دالة للحصول على الإحداثي السيني للنقطة المرجعية / getx

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

**m=getx();**

### دالة للحصول على الإحداثي الصادي للنقطة المرجعية / gety

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

**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 أفقياً مسافة قدرها dx وتبعد مسافة قدرها dy رأسياً) 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*

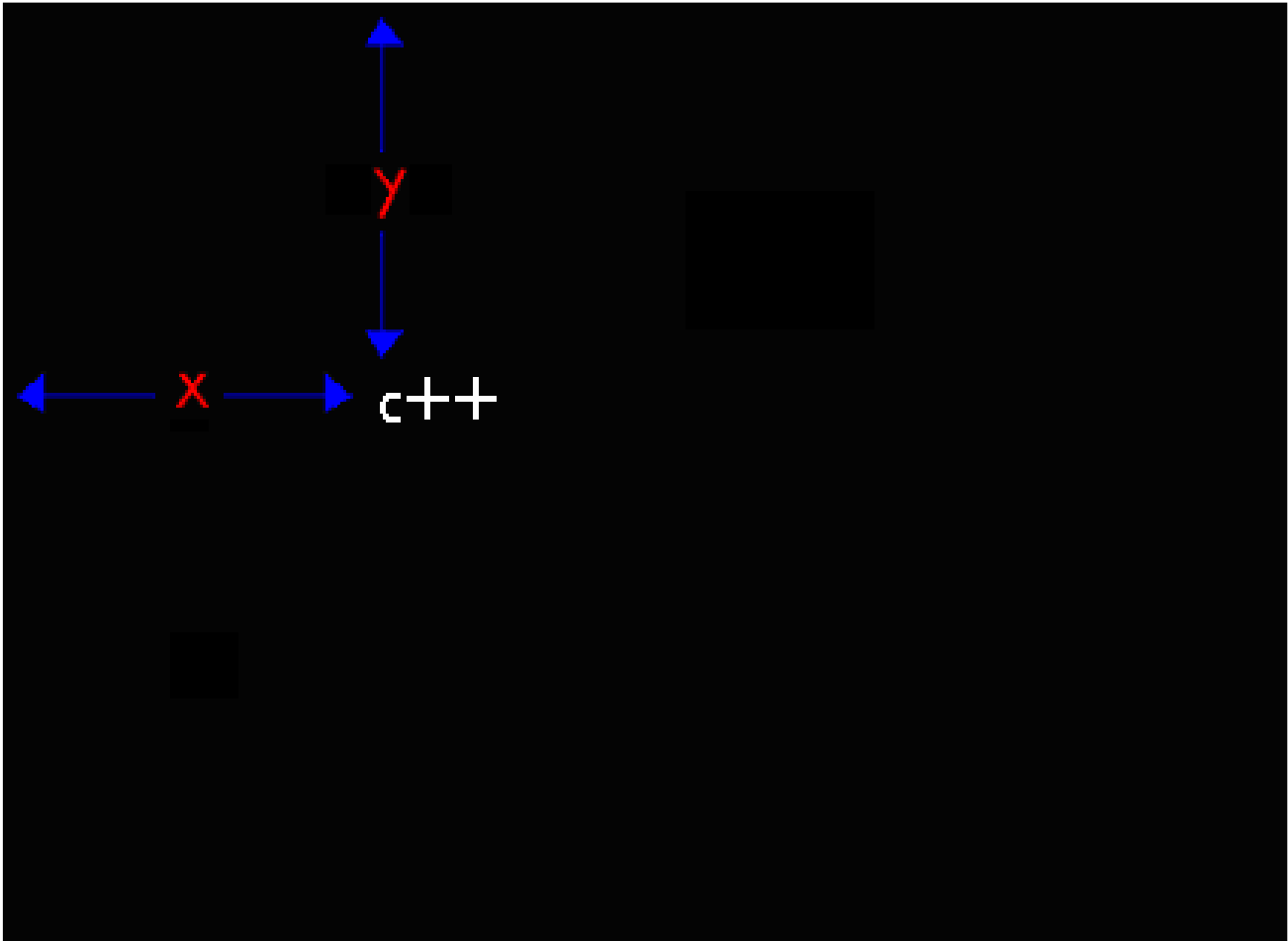


## دالة لإظهار نص عند النقطة المرجعية/ outtext

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

ملاحظة: الموقع الابتدائي 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

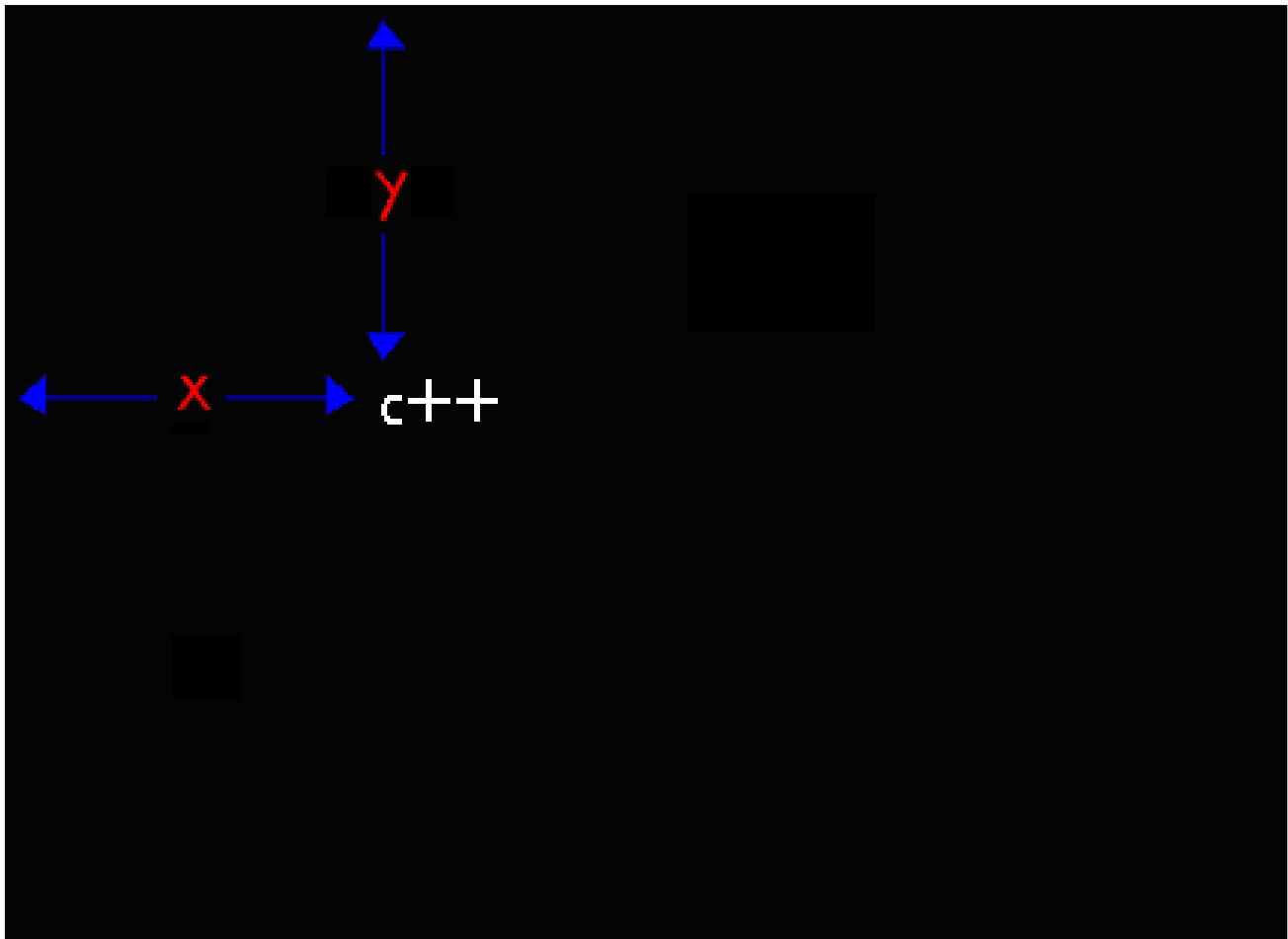
<code>#include&lt;stdio.h&gt;</code>	
<code>#include&lt;conio.h&gt;</code>	
<code>#include&lt;graphics.h&gt;</code>	
<code>void main()</code>	
<code>{</code>	
<code>int x=45,y=60,h=15;</code>	
<code>int gdriver = DETECT, gmode, errorcode;</code>	
<code>initgraph(&amp;gdriver, &amp;gmode, "c:\\tc\\bgi");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

## 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

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

## 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);`**

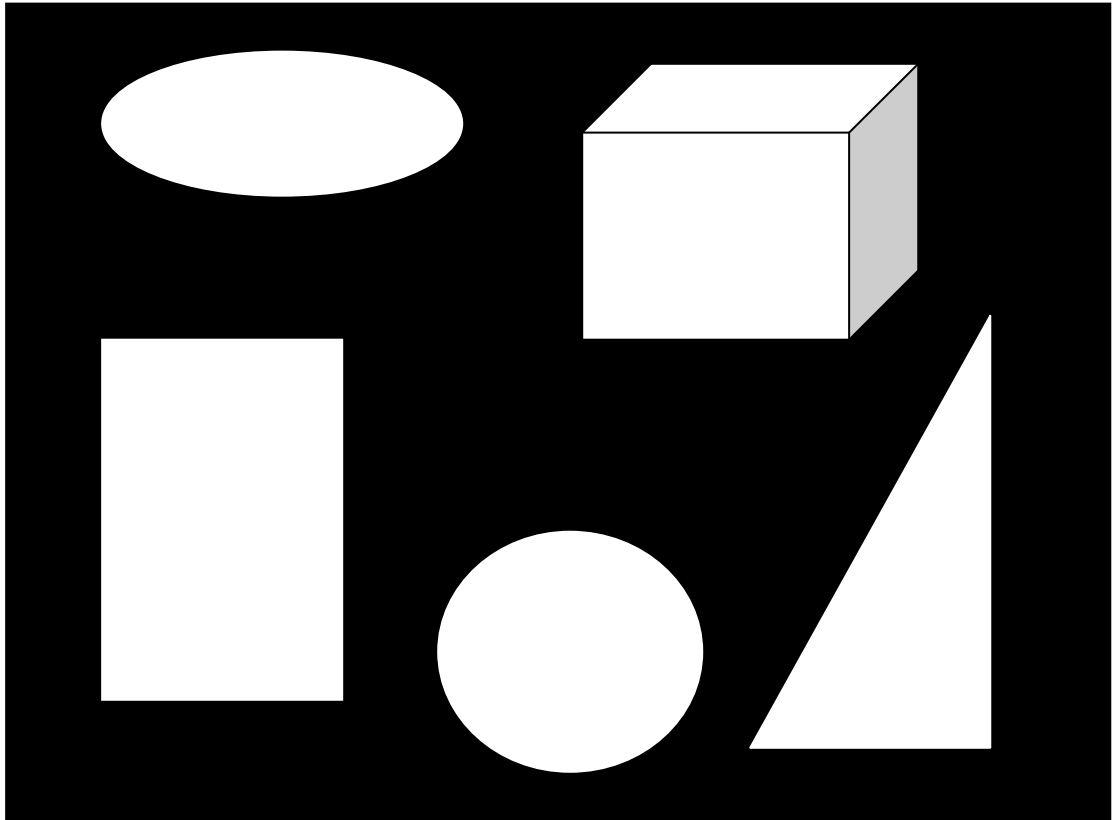
a متغير صحيح.

b متغير صحيح.

msg مصفوفة من نوع char

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

# الفصل السادس / الأشكال الهندسية المسطحة

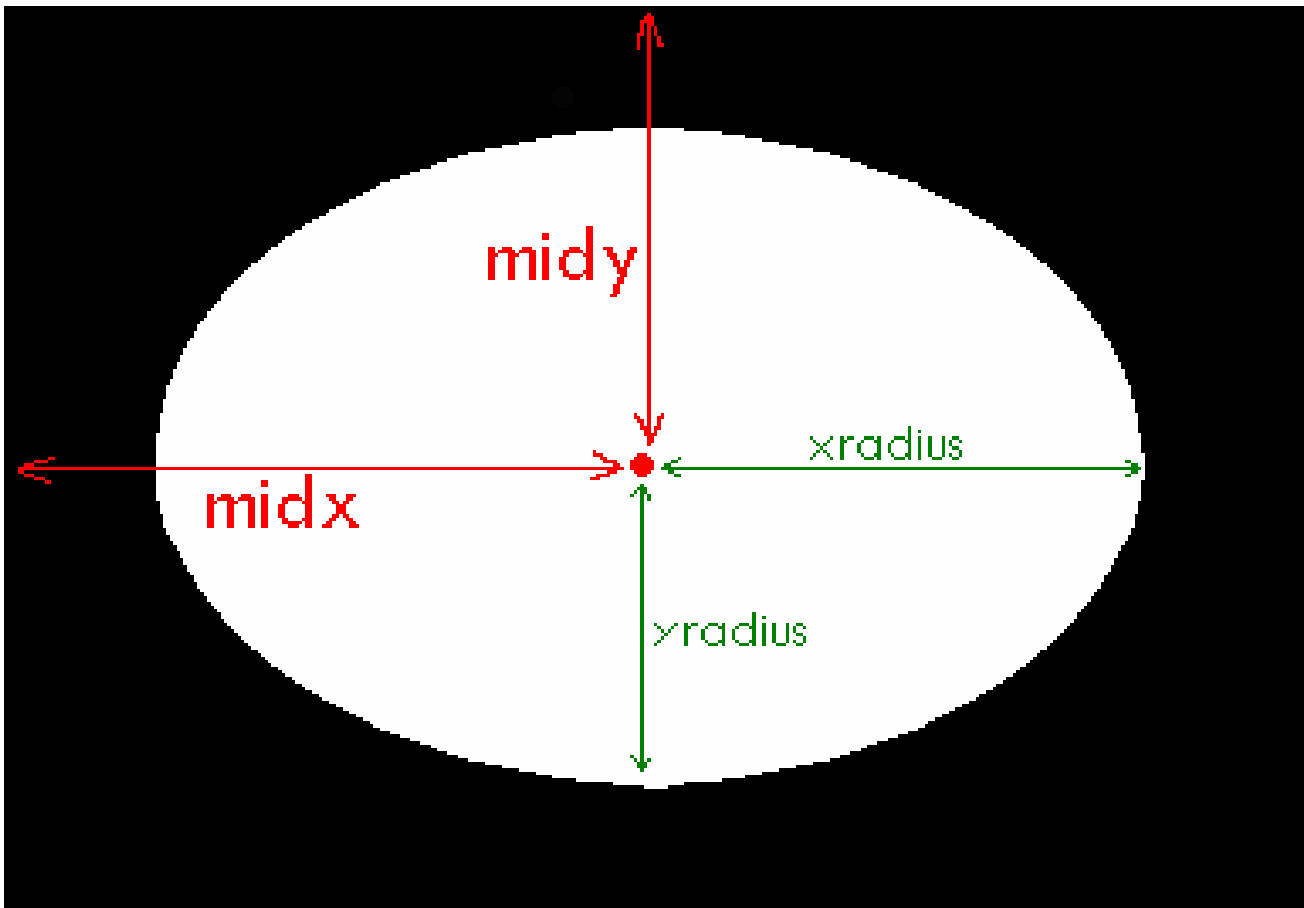


## دالة رسم قطع ناقص مصمت / fillellipse

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);**



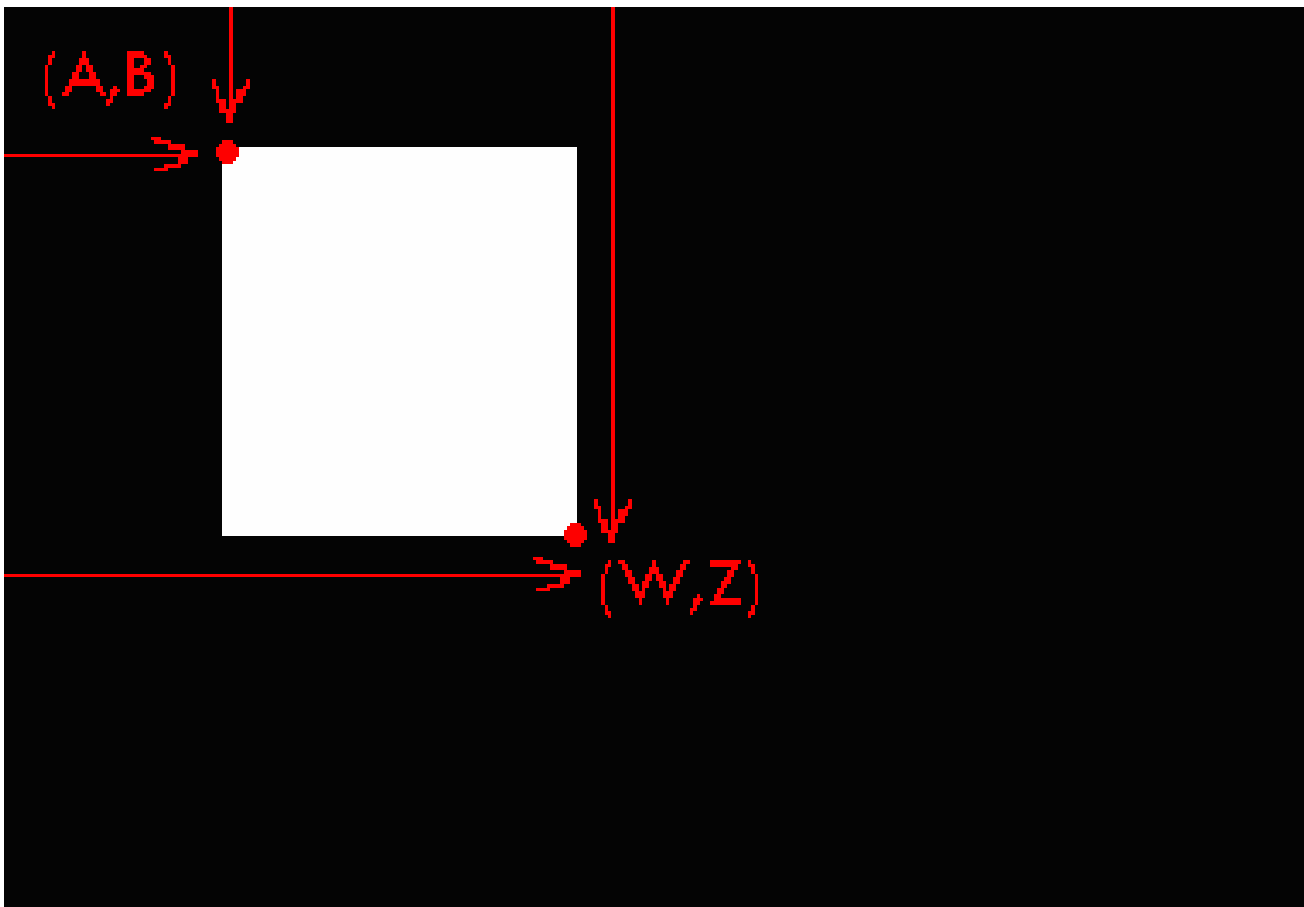
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

#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");	
<b>fillellipse (midx, midy, xradius,yradius);</b>	
getch();	
}	

## دالة رسم مستطيل مصمت / bar

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

**bar(A,B,W,Z);**



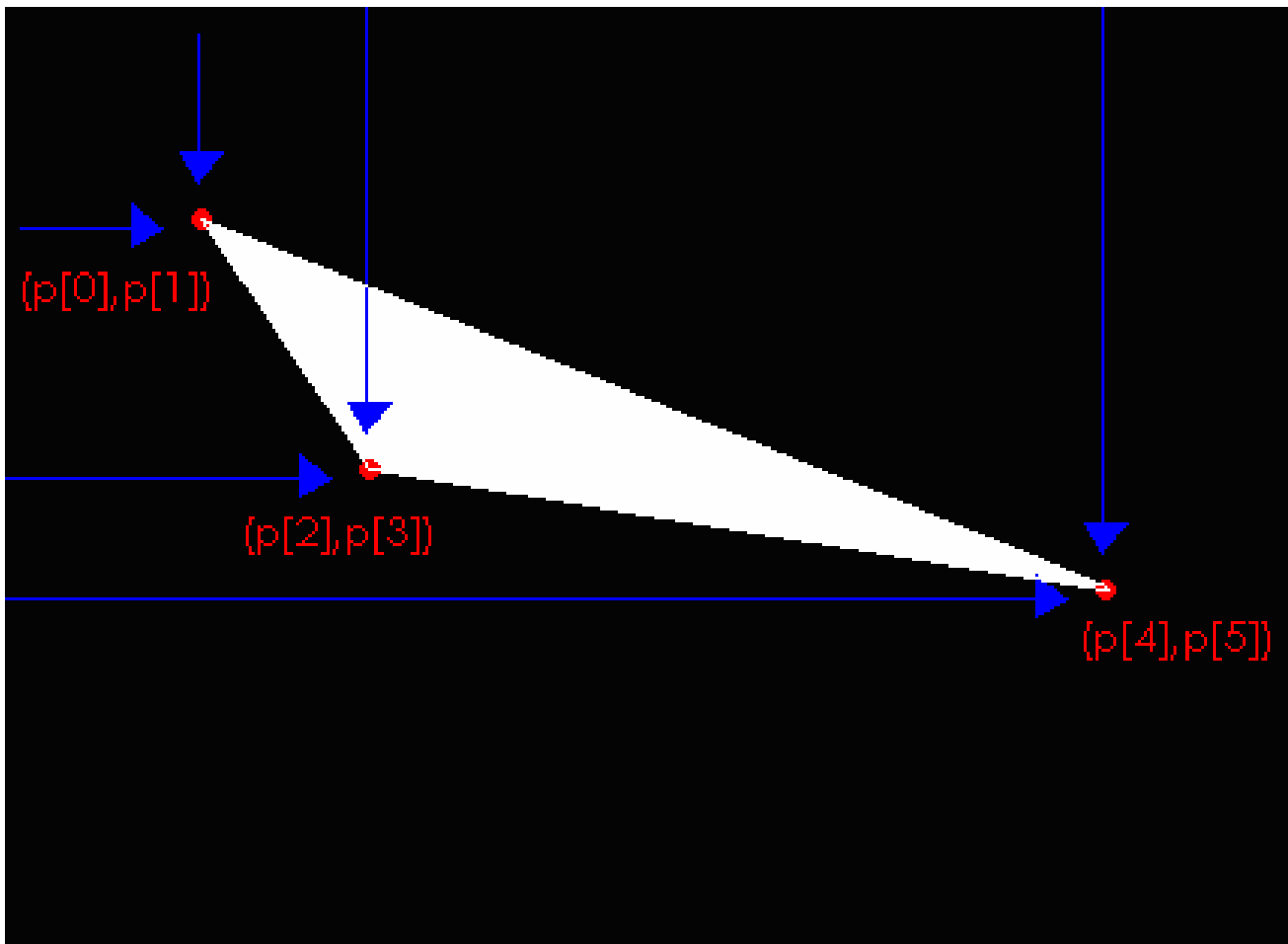
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

#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");	
<b>bar(a,b,w,z);</b>	
getch();	
}	

## دالة رسم أشكال مضلعة مصمته / fillpoly

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

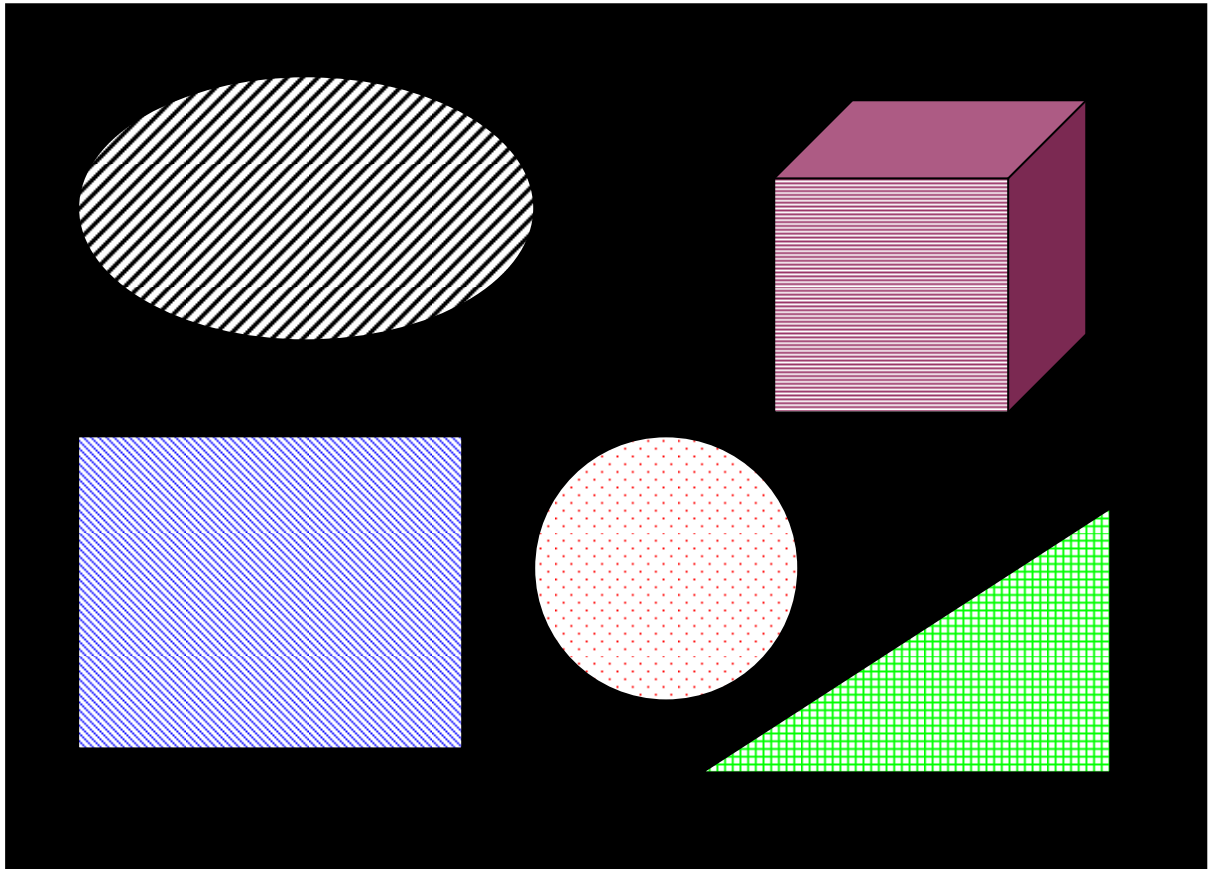
**Fillpoly(n,p);**



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



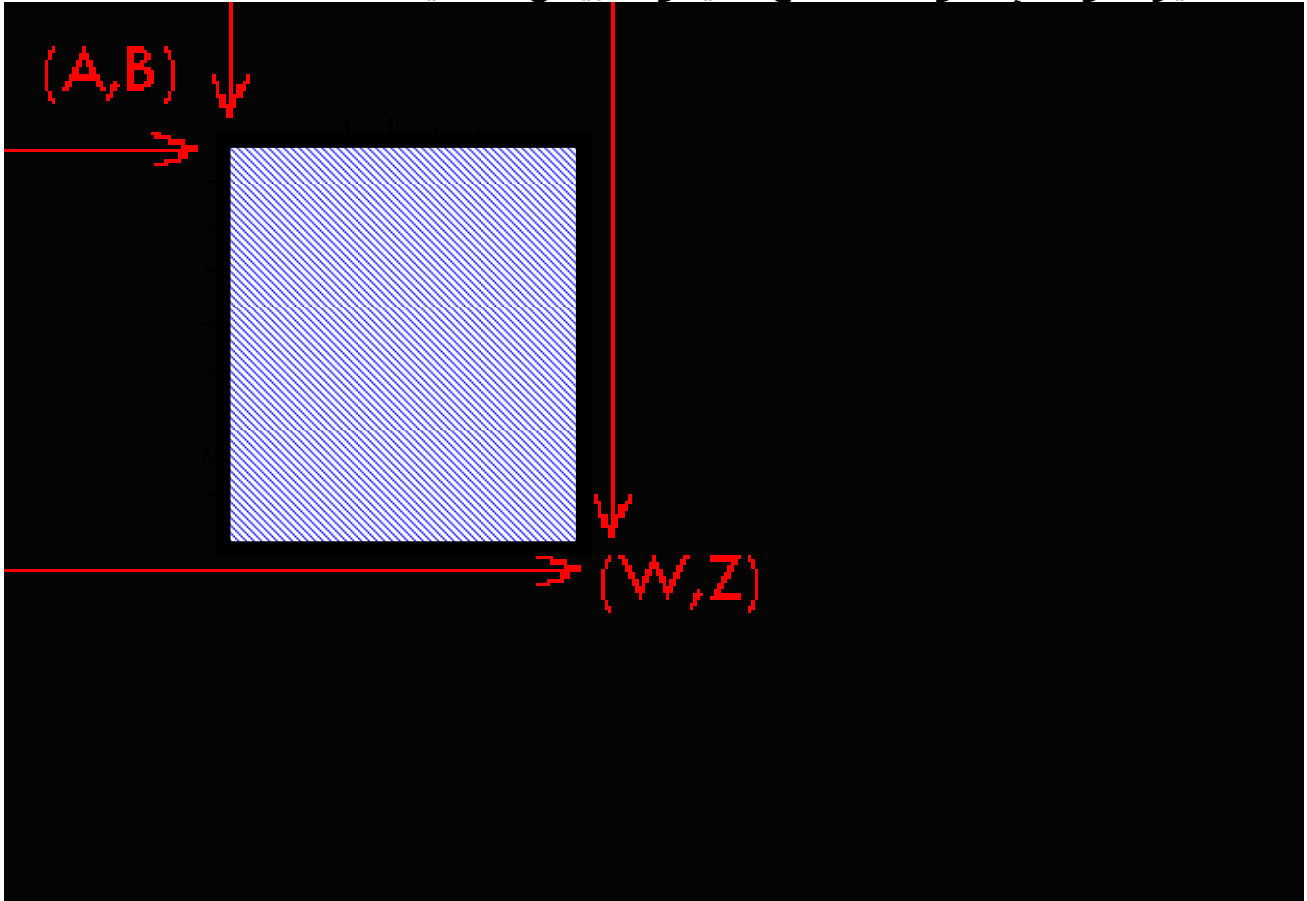


## دالة تغيير نوع ولون السطح المصمت / `setfillstyle`

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);`**

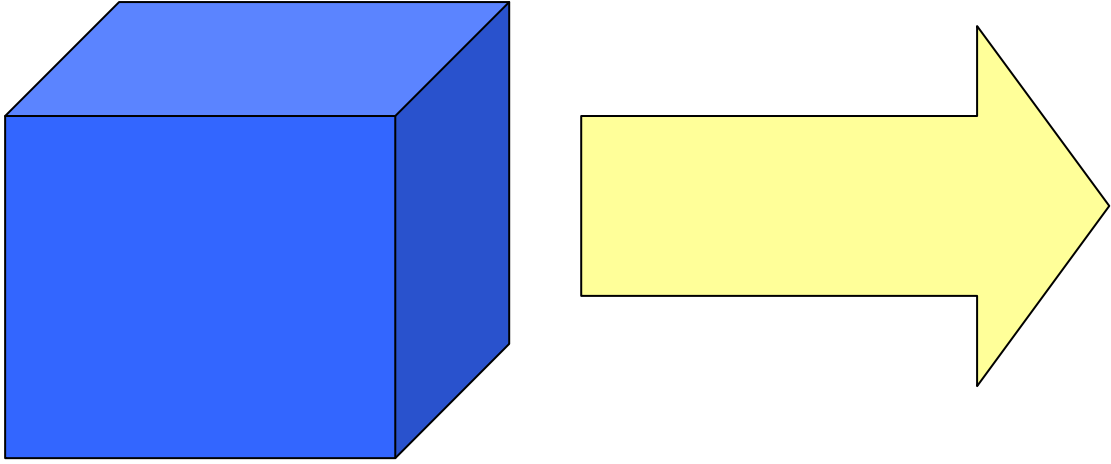
ملاحظة: يجب أن تكتب (هذه الدالة) قبل (دالة الرسم)، وإذا لم تستعمل دالة تغيير اللون فإن لون السطح سيكون أبيض تلقائياً.



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

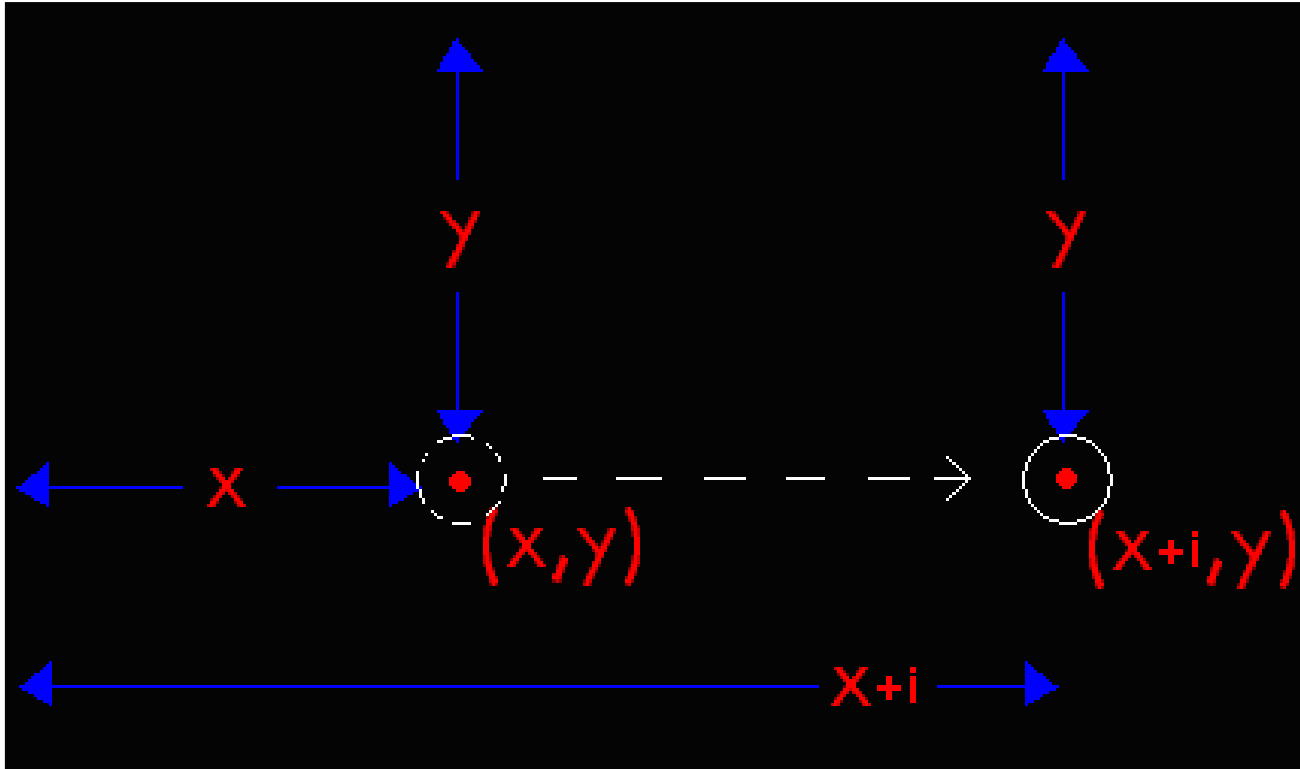
<code>#include&lt;stdio.h&gt;</code>	
<code>#include&lt;conio.h&gt;</code>	
<code>#include&lt;graphics.h&gt;</code>	
<code>void main()</code>	
<code>{</code>	
<code>  Int a=10,b=20,w=150,z=200;</code>	
<code>  int gdriver = DETECT, gmode, errorcode;</code>	
<code>  initgraph(&amp;gdriver, &amp;gmode,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

# الفصل الثامن / طرق الإزاحة



## تصميم برنامج لتحريك دائرة أفقياً

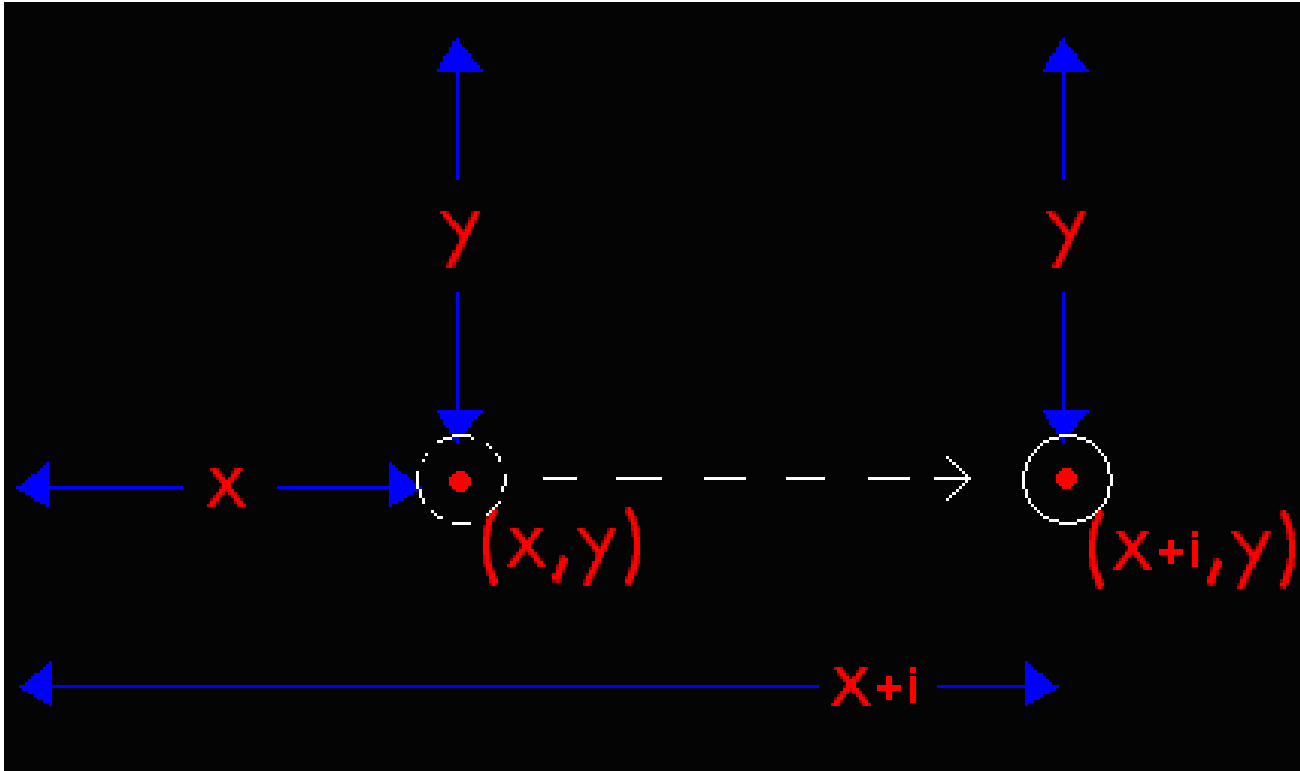
- إذا أردنا تحريك نقطة أفقياً جهة اليمين من  $(x_1, y)$  إلى  $(x_2, y)$ .
- 1. فإننا نرسم الدائرة في  $(x_1, y)$ ، ونصف قطرها  $h$
- 2. ثم نثبتها على الشاشة لحظة، وذلك باستخدام دالة `delay(100)` وهي تقوم بتثبيت الشاشة لمدة  $10\text{ms}$  حيث (الثانية الواحدة =  $1000\text{ms}$ )، ويكتب مقدار الزمن بين قوسي الدالة في المكان المفضل.
- 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.



<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; #include&lt;dos.h&gt; #include&lt;graphics.h&gt;</pre>	مكتبة الدالة <code>delay(100)</code>
<pre>void main() { int x=45,y=60,h=3; int gdriver = DETECT, gmode, errorcode; initgraph(&amp;gdriver, &amp;gmode, "c:\\tc\\bgi"); for(int i=0;i&lt;100;i++) { Setcolor(0); circle(x,y,h); X=x+1 Setcolor(15); circle(x,y,h); delay(100); } getch(); }</pre>	
	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

## طريقة أخرى لتحريك دائرة أفقياً

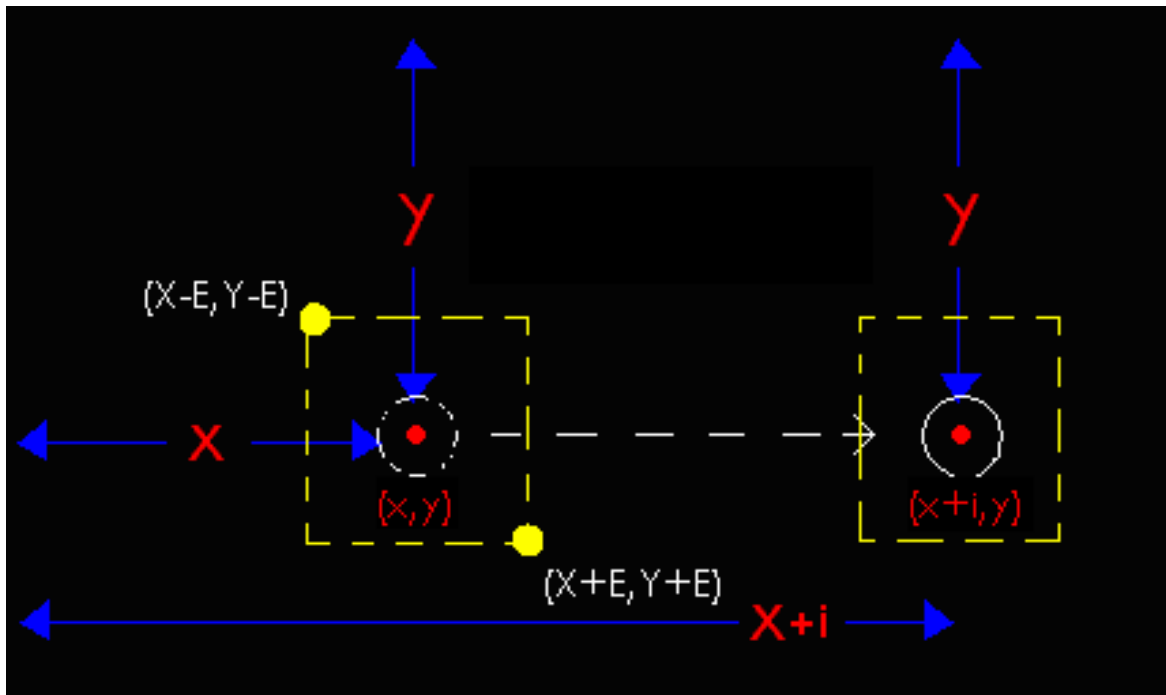
- إذا أردنا تحريك نقطة أفقياً جهة اليمين من  $(x1,y)$  إلى  $(x2,y)$ .
  1. نرسم الدائرة وليكن مركزها هو  $(x1,y)$  ونصف قطرها  $h$
  2. ثم نثبتها على الشاشة لحظة، وذلك باستخدام دالة `delay(100)` وهي تقوم بتهيئة الشاشة لمدة  $10ms$  (الثانية الواحدة =  $1000ms$ )، ويكتب مقدار الزمن بين قوسي الدالة في المكان المطلوب.
  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.



<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; #include&lt;dos.h&gt; #include&lt;graphics.h&gt;</pre>	مكتبة الدالة <code>delay(100);</code>
<pre>void main() { int x=45,y=60,h=3; int gdriver = DETECT, gmode, errorcode; initgraph(&amp;gdriver, &amp;gmode, "c:\\tc\\bgi");</pre>	
<pre>for(int i=0;i&lt;100;i++) { cleardevice (); x+=1; circle(x,y,h); delay(100); }</pre>	<p>مقدار الإزاحة من 0 إلى 100 نقطة.</p> <p>دالة مسح الشاشة إزاحة أفقية 1</p> <p>دالة تهيئة الشاشة لمدة 100 مللي ثانية</p>
<pre>getch(); }</pre>	

## تصميم برنامج لتحريك دائرة أفقياً (إيقاف 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);
}
```