

برمجة الرسم بلغة سي المحسنة

الجزء الأول

TURBO C PLUS PLUS 3.0

باستخدام



برمجة: البراء عبد الرؤوف الرملي

طرابلس / ليبيا

نسخة © 2007 , حقوق الطبع محفوظة SBR



Software Bara Ramli (SBR)

لا يسمح بإعادة طبع هذا الكتاب إلا بإذن خطي مسبق
من المؤلف.

بينما يسمح بنسخه و تنويره في نطاق الاستعمال
الشخصي (الغير تجاري) , ولكن لا يمكنك الادعاء بأنك
من قام بهذا العمل وعليك الإشارة لمؤلفه الأصلي.

ملاحظة: يقدم هذا الكتاب كما هو من دون أي
كفالة أو ضمان لمتوياته.

All programs in this book is free software:

you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

مقدمة

أقدم إليكم كتاب برمجة الرسم ويحتوي على أساسيات الرسم على الشاشة مع شرح لدوال مكتبة الرسم graphics.h

أرجو الله أن ينفع به وأن يكون مساهمة منا في إثراء المكتبة العربية والمبرمج العربي.

ملاحظة: المكتبات والبرامج المعروفة , مرفقة مع الكتاب في مجلد (المرفقات).

وأريد أن أنبه على أن البرامج والمكتبات المعروفة في هذه السلسلة , لا زالت تحتاج إلى تطوير وإضافات, وهذا يقع على عاتقنا جميعا حتى نصل بها إلى المستوى المطلوب , إذا فهمي الآن بين يديك لتخيف إليها ما تظن أنه يرقى بها إلى الأفضل ومن ثم تقوم بنشرها لنعلم الفائدة لنا جميعا , لأنه ما لم نتشارك بأفكارنا , فلن نتقدم خطوة إلى الأمام.

البراء عبد الرؤوف الرملي
opencpp@yahoo.com
طرابلس/ليبيا

المحتويات

الفهرس

مفاهيم أساسية

الفصل الأول/ دوال الرسم الموجودة في مكتبة graphics.h

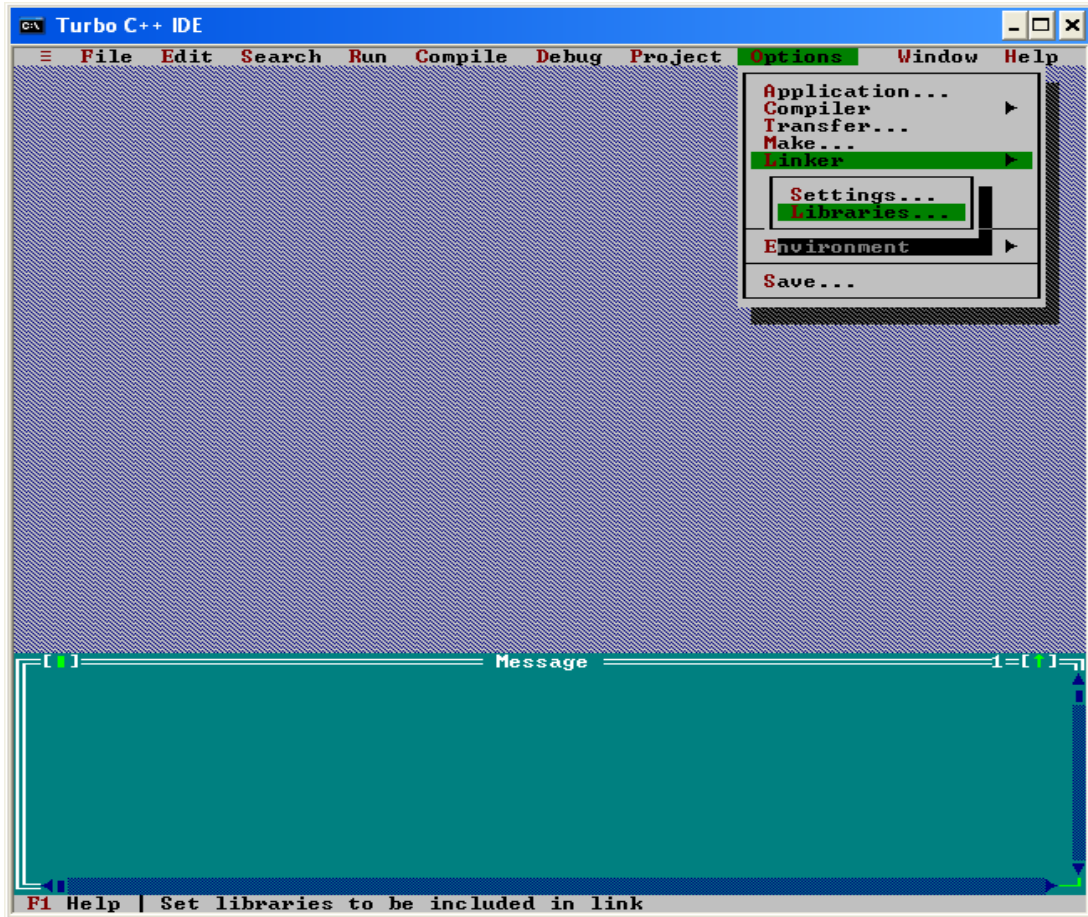
الفصل الثاني/ الأشكال الهندسية

الفصل الثالث/ تلوين الأشكال الهندسية

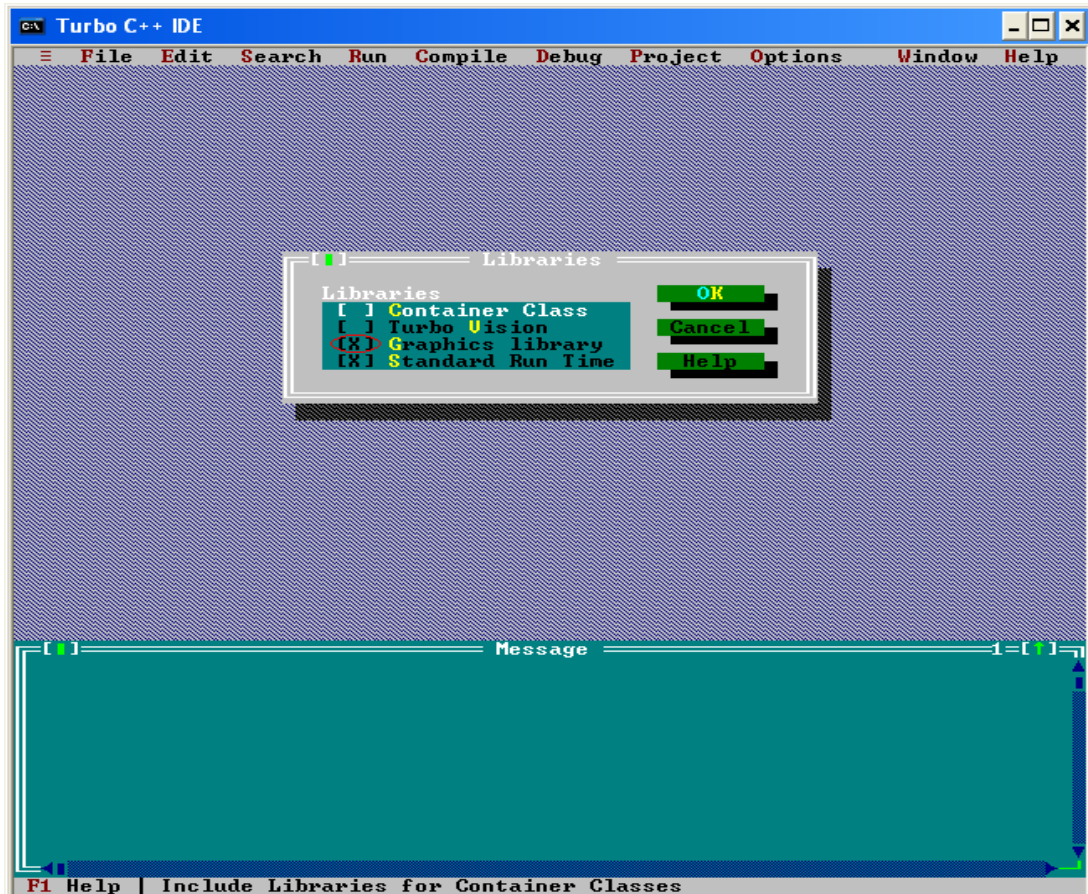
مفاهيم أساسية

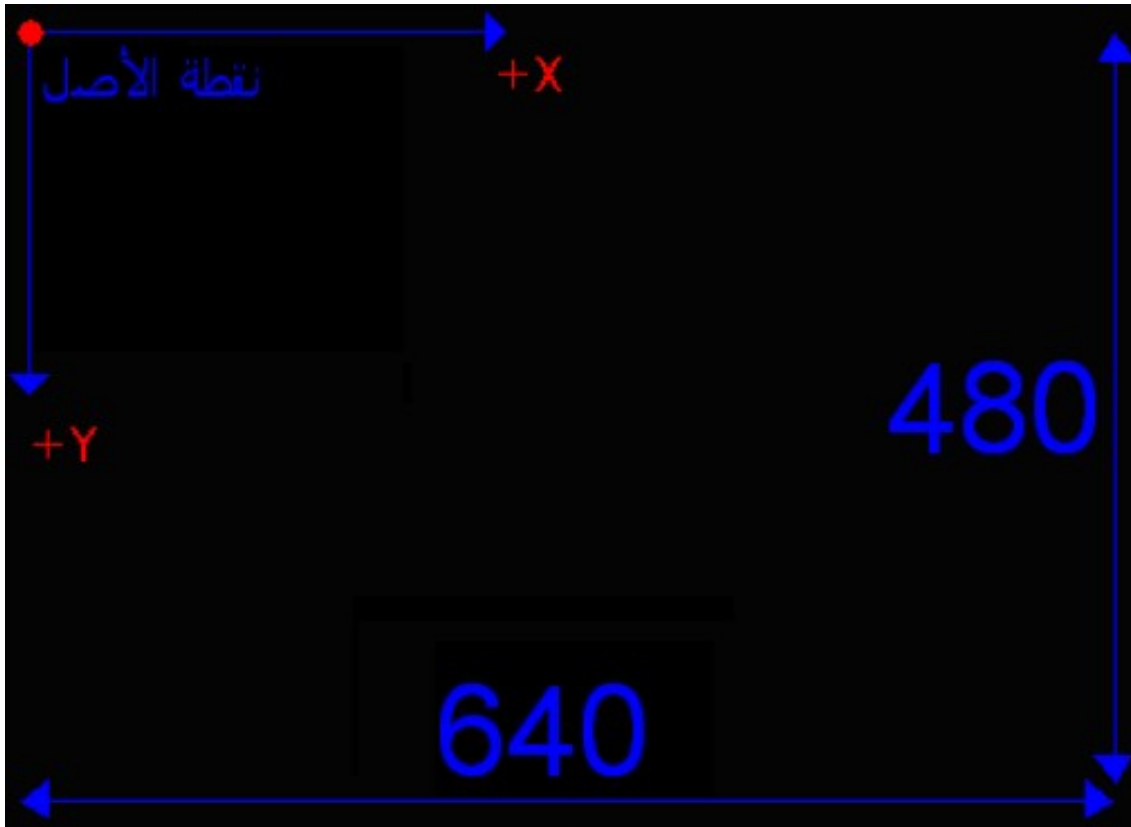


اضغط على Options ثم Linker ثم Libraries



اختر Graphics library ثم اضغط ok





يجب أن تعلم:

1. أن الشاشة قسمت أفقيا إلى 640 نقطة (pixel) ورأسيا إلى 480 نقطة (pixel), وتكون دقة الصورة = 640×480
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 فإن الرسم قد يخرج عن حدود الشاشة.

الفصل الأول /

دوال الرسم في مكتبة graphics



الشكل العام لبرنامج الرسم

يجب كتابة النصوص الملونة بالأحمر في كل برنامج رسم:

<pre>#include <stdio.h> #include <conio.h> #include <graphics.h> void main() { int gdriver = DETECT, gmode, errorcode; initgraph(&gdriver, &gmode, "c:\\tc\\bgi");</pre>	استدعاء مكتبة الرسم
	تؤدي للتعرف على كارت الرسم الموجود
	بمعنى 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.
<pre>..... (دوال الرسم تكتب هنا)</pre>	
<pre>getch(); }</pre>	تثبيت الصورة على الشاشة لعين الضغط على أي زر

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

الدوال الموجودة في الكتاب (بجزئيه الأول والثاني) مظللة 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) وتخصصه للمتغير الصحيح `m` مثلاً.

`m=getmaxx();`

دالة للحصول على أقصى إحداثي صادي للشاشة / `getmaxy`

ترجع الدالة أقصى قيمة للإحداثي الصادي على الشاشة, وهو يساوي 480.

تنبيه: ترجع الدالة رقم (480) وتخصصه للمتغير الصحيح `m` مثلاً.

`m=getmaxy();`

الفصل الثاني /

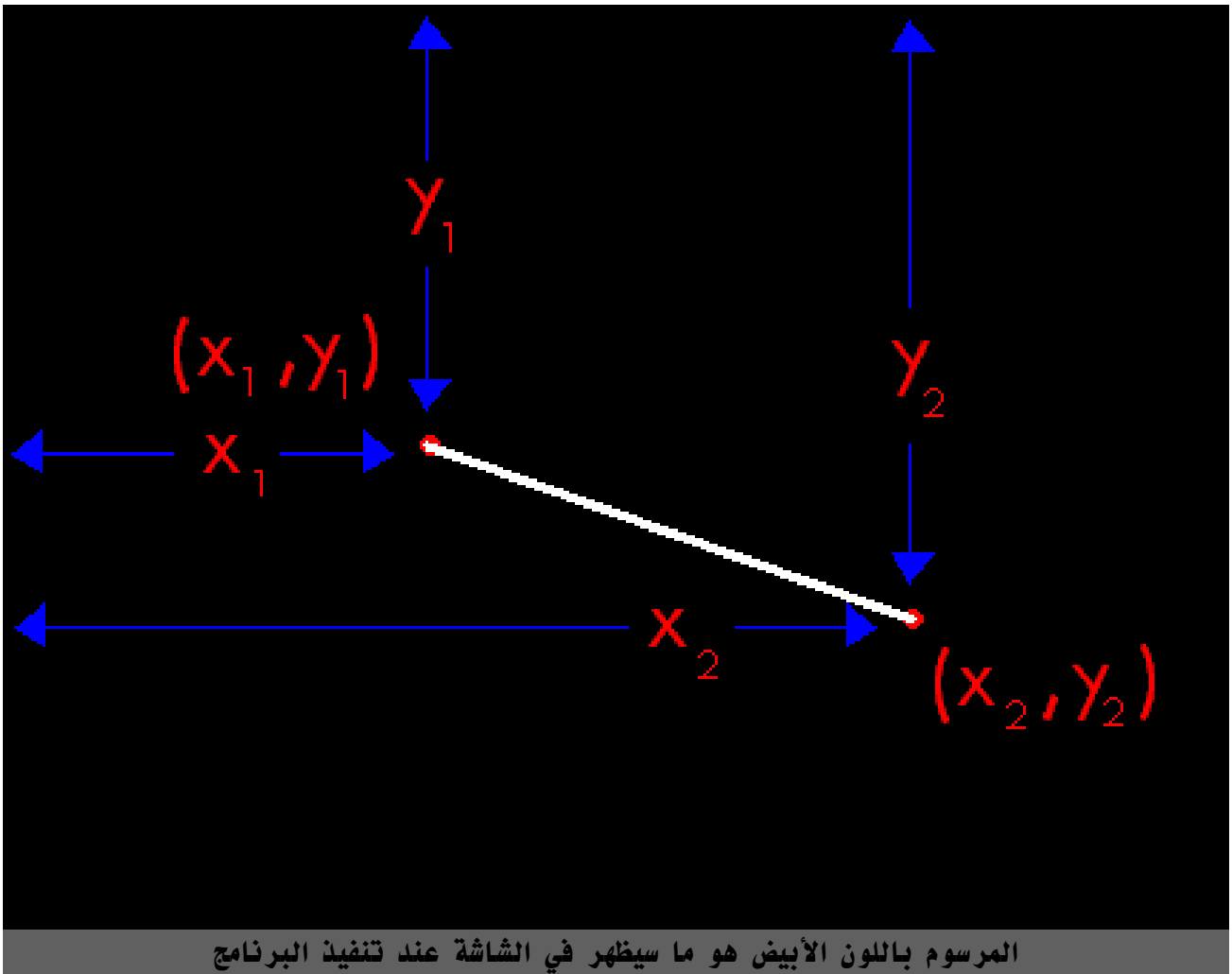
الأشكال الهندسية

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);

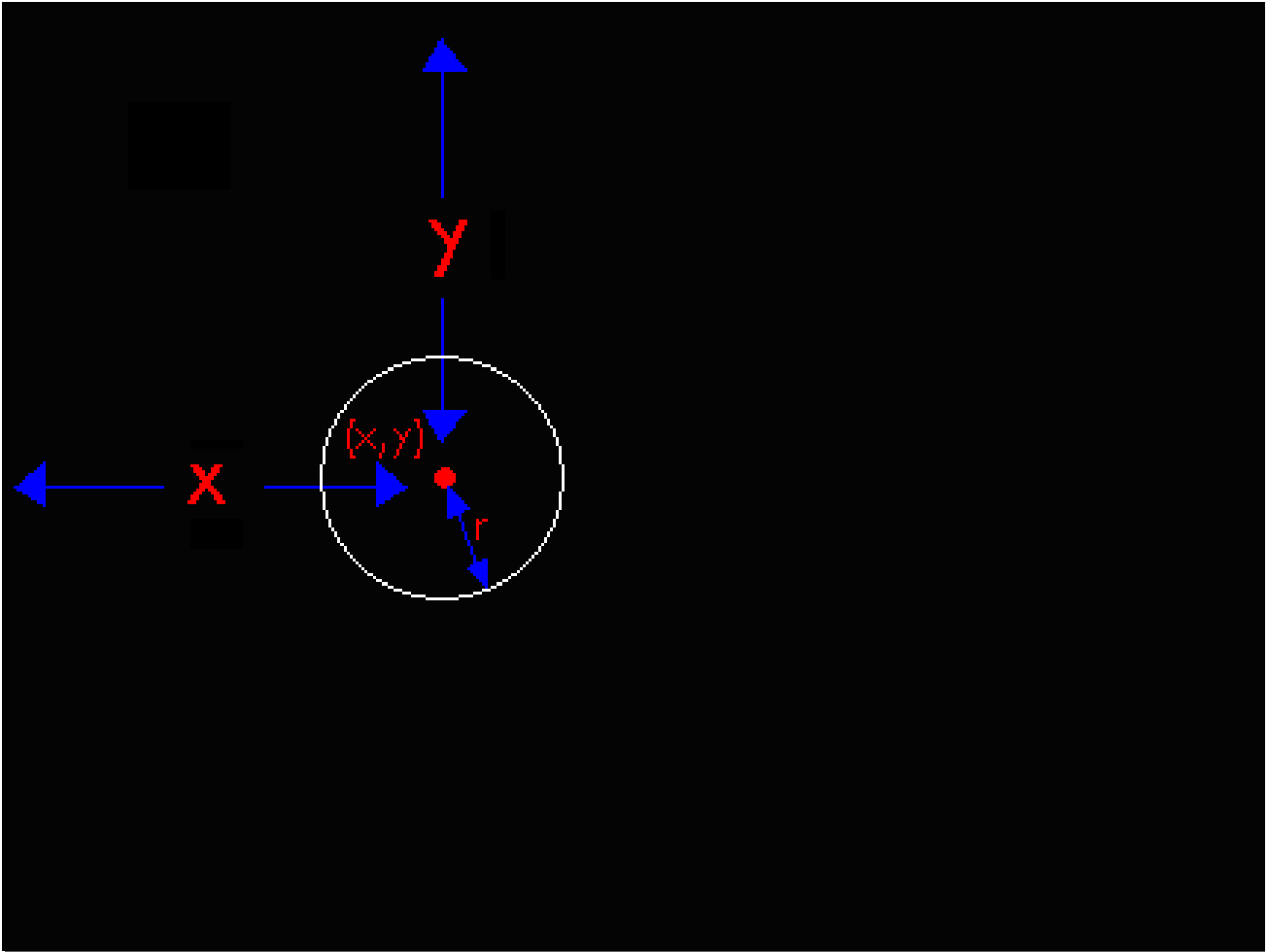


<pre>#include <stdio.h></pre>	
<pre>#include <conio.h></pre>	
<pre>#include <graphics.h></pre>	
<pre>void main()</pre>	
<pre>{</pre>	
<pre>int x1=50 ,y1=20;</pre>	
<pre>int x2=200,y2=100;</pre>	
<pre>int gdriver = DETECT, gmode, errorcode;</pre>	
<pre>initgraph(&gdriver, &gmode, "c:\\tc\\bgi");</pre>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

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);



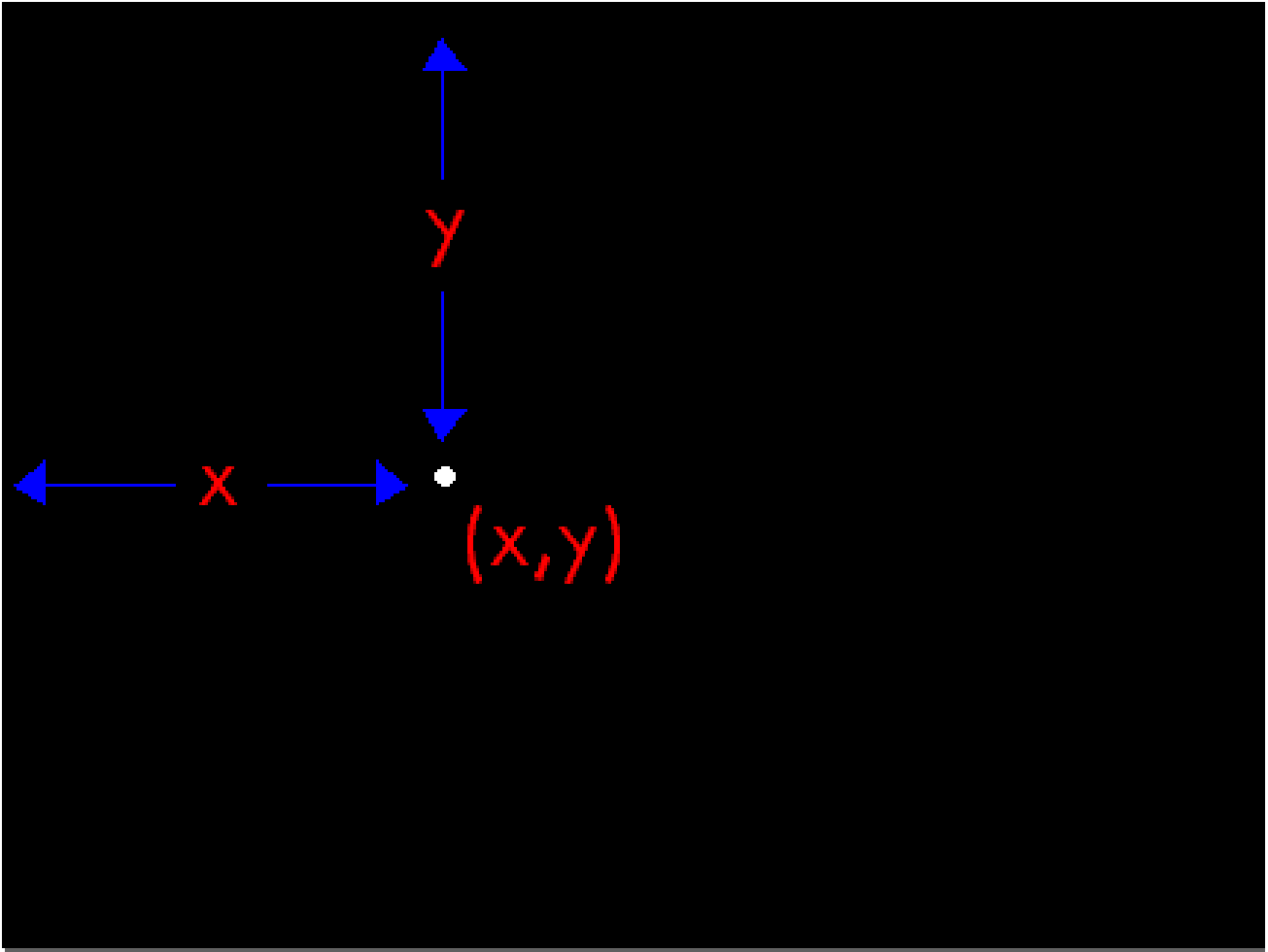
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

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

دالة رسم نقطة / putpixel

تحتوي على 3 متغيرات عددية هم: إحداثي النقطة (x1,y1) ومتغير ثابت 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);



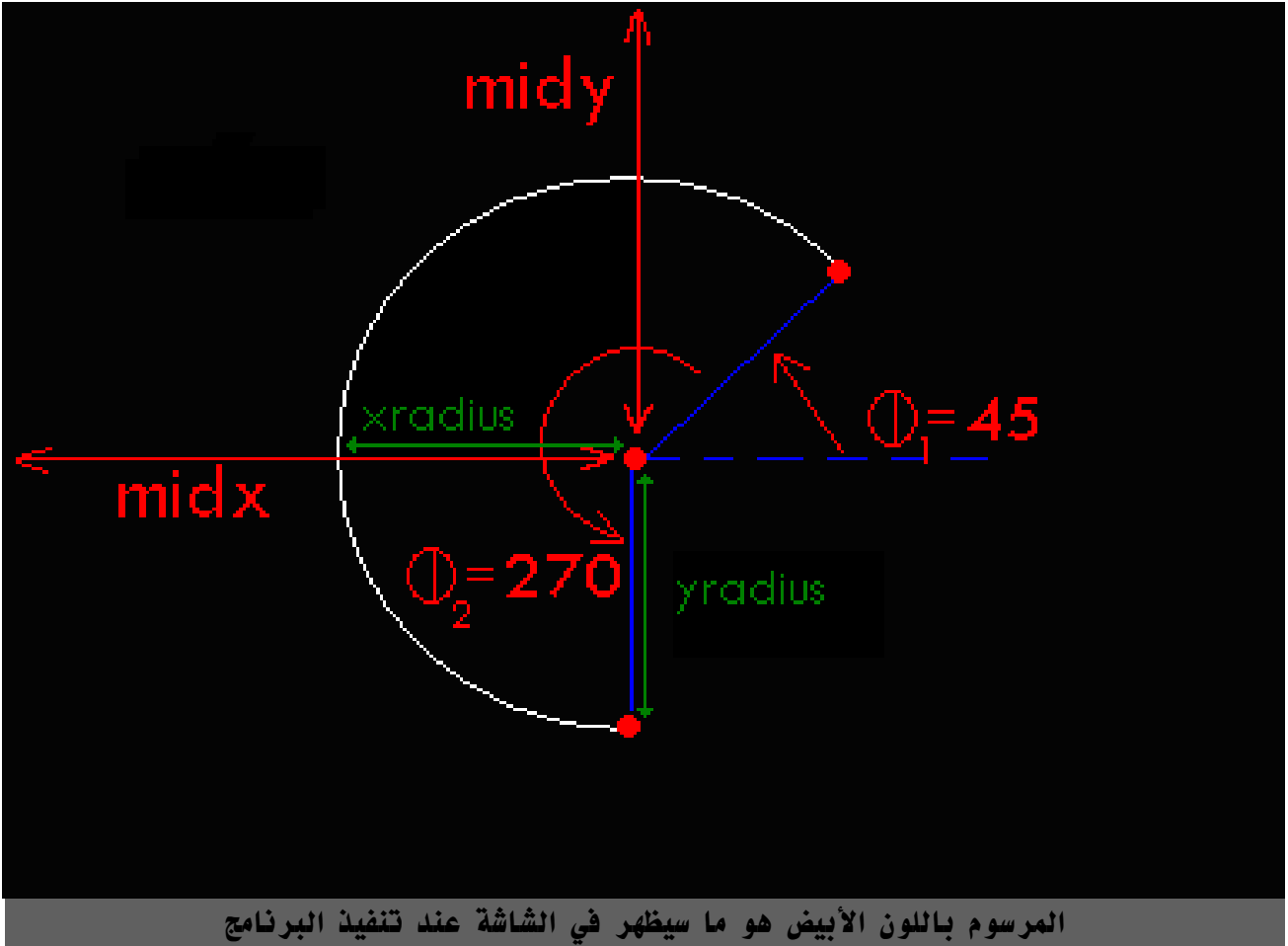
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

<code>#include <stdio.h></code>	
<code>#include <conio.h></code>	
<code>#include <graphics.h></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(&gdriver, &gmode,</code> <code>"c:\\tc\\bgi");</code>	
<code>putpixel(x,y,h);</code>	
<code>getch();</code>	
<code>}</code>	

دالة رسم أقواس دائرية / arc

(midx, midy) = إحداثي المركز
stangle = (Φ_1) زاوية البدء
endangle = (Φ_2) زاوية النهاية
radius = نصف القطر

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



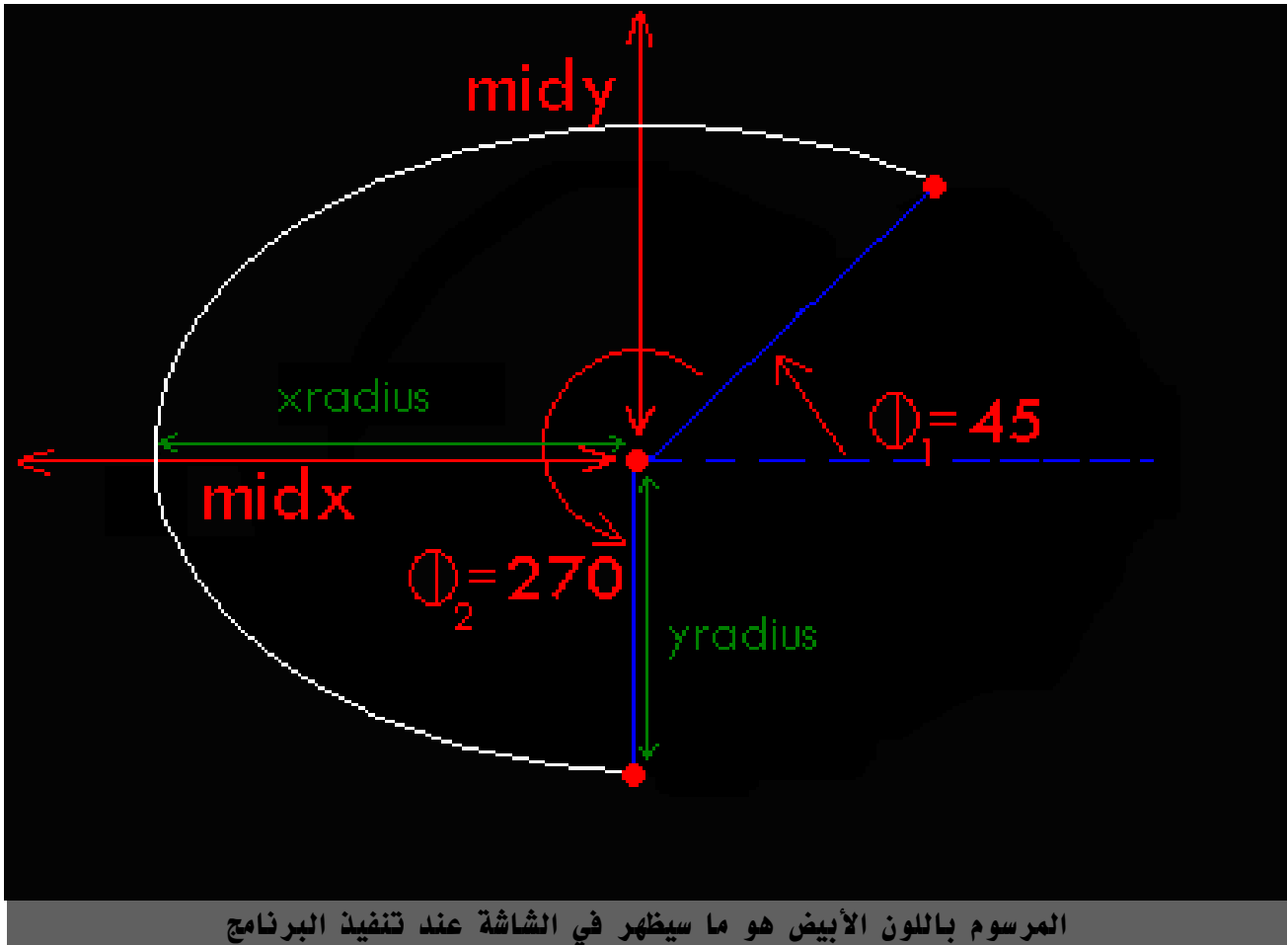
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

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

دالة رسم قطع ناقص / ellipse

(midx, midy) = إحداثي المركز
stangle = (Φ_1) زاوية البدء
endangle = (Φ_2) زاوية النهاية
xradius = x نصف قطر الإحداثي
yradius = y نصف قطر الإحداثي

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



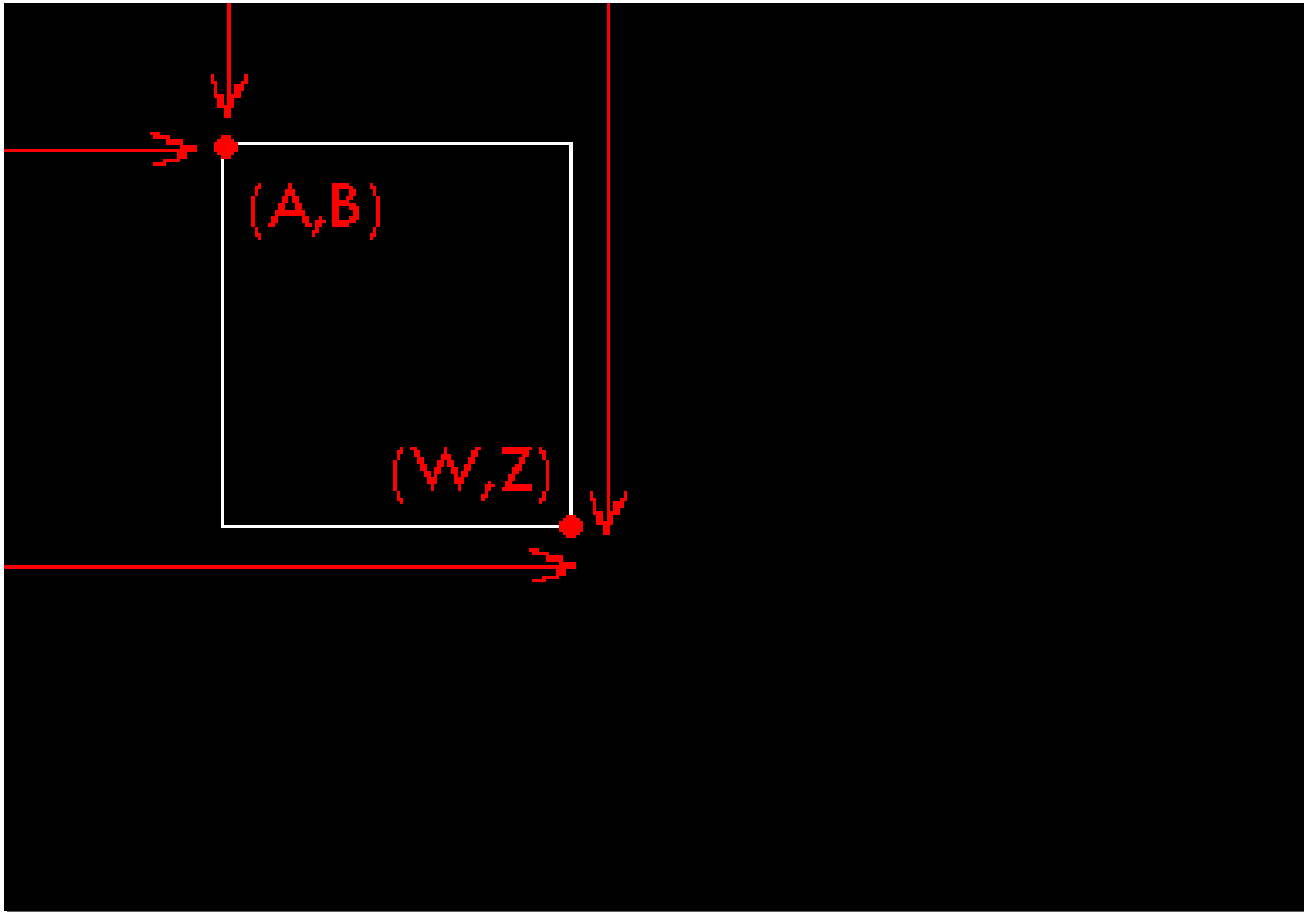
<pre>#include <stdio.h></pre>	
<pre>#include <conio.h></pre>	
<pre>#include <graphics.h></pre>	
<pre>void main()</pre>	
<pre>{</pre>	
<pre>int midx=320, midy=240, stangle = -45;</pre>	
<pre>int endangle = 180, xradius = 70, yradius = 30;</pre>	
<pre>int gdriver = DETECT, gmode, errorcode;</pre>	
<pre>initgraph(&gdriver, &gmode, "c:\\tc\\bgi");</pre>	
<pre>ellipse (midx, midy, stangle, endangle,xradius,yradius);</pre>	
<pre>getch();</pre>	
<pre>}</pre>	

دالة رسم مستطيل / rectangle

(A,B) = إحداثي الركن الأيسر

(W,Z) = إحداثي الركن الأيمن

rectangle(A,B,W,Z);



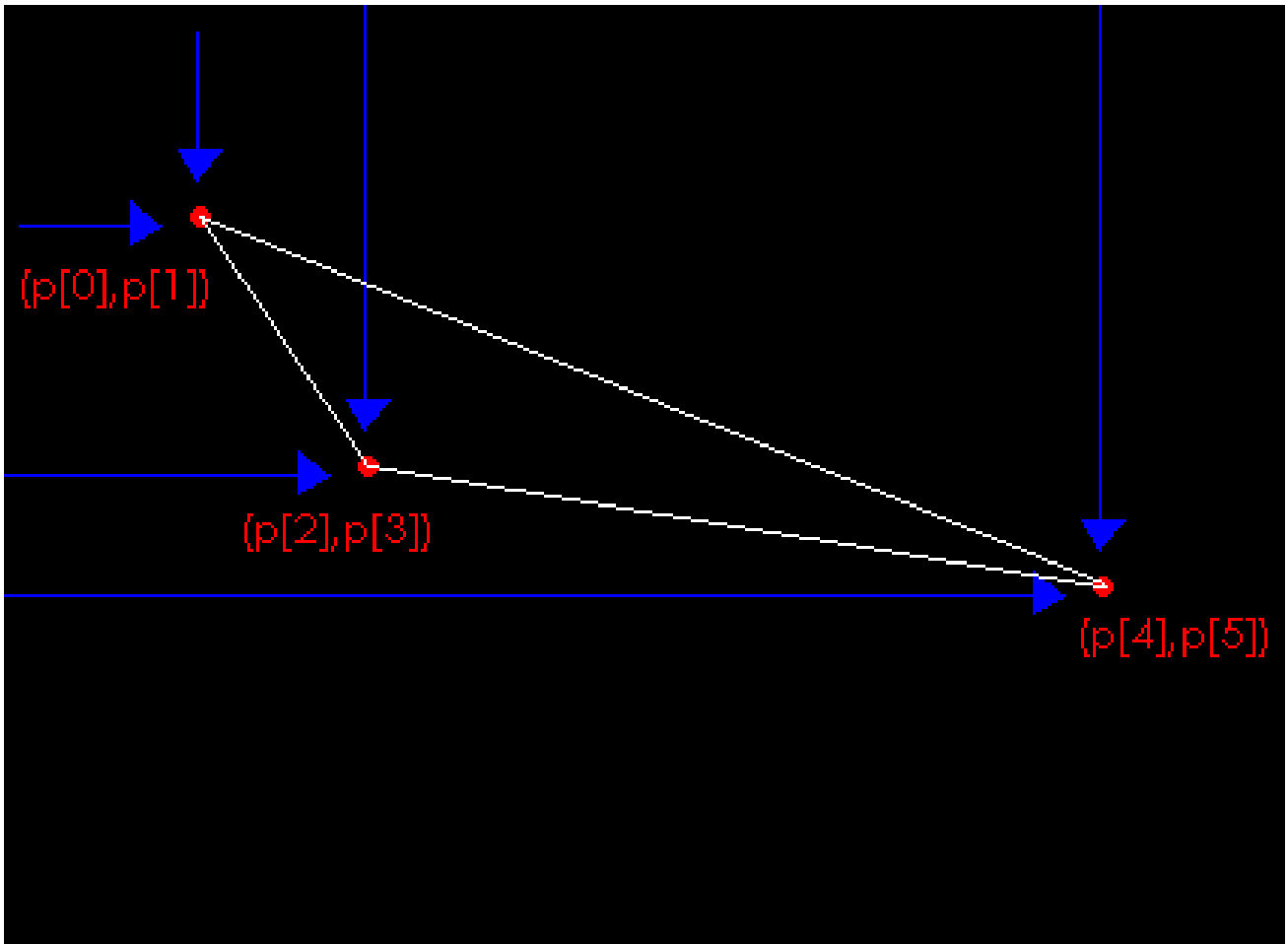
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

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

دالة رسم الأشكال المضلعة / drawpoly

اسم مصفوفة النقاط هي **p** وتحتوي على عدد **n** من النقاط.

drawpoly(n,p);



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

<pre>#include <stdio.h></pre>	
<pre>#include <conio.h></pre>	
<pre>#include <graphics.h></pre>	
<pre>void main()</pre>	
<pre>{</pre>	
<pre>int p[6];</pre>	
<pre>int gdriver = DETECT, gmode, errorcode;</pre>	
<pre>initgraph(&gdriver, &gmode, "c:\\tc\\bgi");</pre>	
<pre>p[0]=10;p[1]=15; p[2]=100;p[3]=200; p[4]=200;p[5]=250; drawpoly(3,p);</pre>	
<pre>getch();</pre>	
<pre>}</pre>	

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 تغيير اللون فإن لون السطح سيكون أبيض تلقائياً.

دالة تغير لون الرسم / Setcolor

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

setbkcolor(4);

ملاحظة: يجب أن تكتب (هذه الدالة) قبل (دالة الرسم), وإذا لم تستعمل دالة تغيير اللون فإن لون الخلفية سيكون أسود تلقائياً.

دالة للحصول على لون الخلفية / getbkcolor

تمكنك 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

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

m=getcolor();

دالة لتلوين الأشكال الهندسية/ floodfill وأخرى لتعريف لون ونوع التعبئة setfillstyle

حيث A هو لون التعبئة بينما SOILD_FILL فنوع التعبئة " وفي هذا النوع مصمته".

setfillstyle(SOLID_FILL,A);

تنبيه: يجب رسم الشكل المراد تعبئته بعد هذه الدالة مباشرة.

ومن بعد ذلك نقوم بصب اللون داخل الشطل باستخدام الدالة floodfill
لصب اللون داخل شكل مغلق, حيث (G,F) إحداثيات نقطة تقع داخل الشكل المغلق.
بينما C هو لون حدود الشكل المراد صب اللون بداخله.

floodfill(G,F,C);

تستعمل هذه الدالة لتلوين الأشكال الهندسية المغلقة مثل: الدائرة والمثلث والمربع والمستطيل....
ويؤدي استعمالها مع غير الأشكال المغلقة إلى تلوين الشاشة بالكامل.

تنبيه: يجب أن تكتب (هذه الدالة) بعد (دالة الرسم).

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();

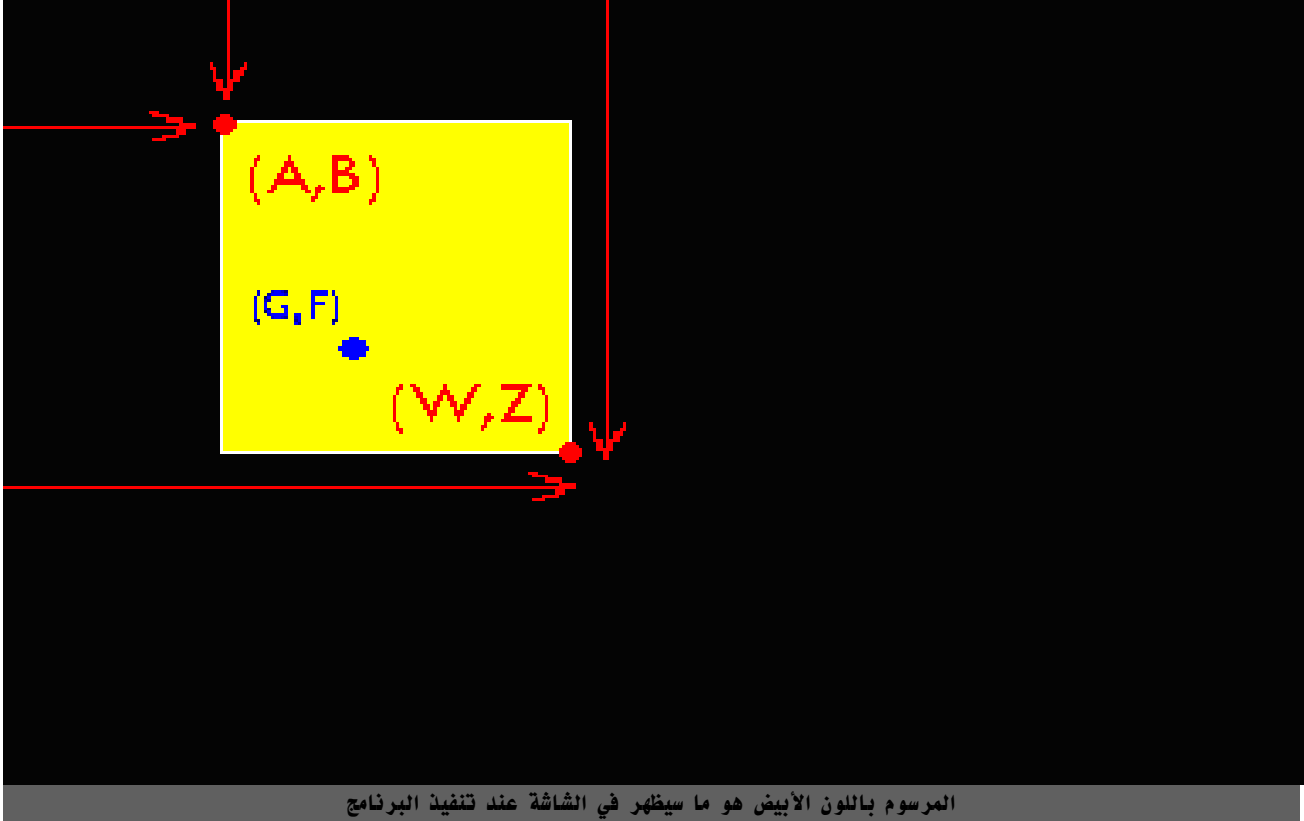
مثال: برنامج لرسم مستطيل وتلوينه.

حيث أركان المستطيل:

(A,B) = إحداثي الركن الأيسر للمستطيل

(W,Z) = إحداثي الركن الأيمن للمستطيل

(G,F) = بينما إحداثي نقطة داخل المستطيل



<pre>#include <stdio.h></pre>	
<pre>#include <conio.h></pre>	
<pre>#include <graphics.h></pre>	
<pre>void main()</pre>	
<pre>{</pre>	
<pre>int</pre>	
<pre>a=100,b=200,w=200,z=300,G=150,F=250;</pre>	
<pre>int gdriver = DETECT, gmode, errorcode;</pre>	
<pre>initgraph(&gdriver, &gmode, "c:\\tc\\bgi");</pre>	
<pre>setfillstyle(SOLID_FILL,14);</pre>	
<pre>setcolor(15);</pre>	
<pre>rectangle(a,b,w,z);</pre>	
<pre>floodfill(G,F,15);</pre>	
<pre>getch();</pre>	
<pre>}</pre>	

ينبع الجزء الثاني