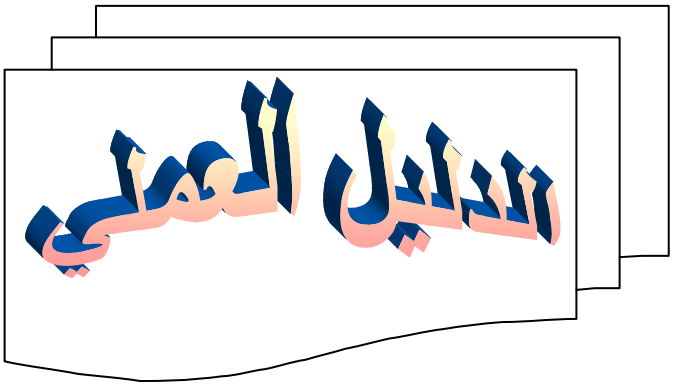
جامعة القدس المفتوحة

كلية التكنولوجيا والعلوم التطبيقية



لمقرر الرسم بالحاسوب-1495

إعداد

أ. طارق رحال

2015-2014

فهرس محتويات الدليل

الأهداف:

ينتظر منك عزيزي الطالب، بعد فراغك من تنفيذ تدريبات هذا الدليل أن تكون قادراً على أن:
تعريف واستخدام الأوامر الخاصة بشاشة الرسم.
تعريف واستخدام أوامر رسم الأشكال الهندسية.
تعريف واستخدام أوامر الكتابة على شاشة الرسم.
تعريف واستخدام أوامر تلوين الأشكال الهندسية.
تعريف واستخدام أوامر تحريك الأشكال.
تعريف واستخدام أوامر الفأرة لاستخدامها في تحديد نقاط الرسم.

مقدمة:

عزيزي الطالب نقدم بين يديك الدليل العملي لمقرر الرسم بالحاسوب رقم 1495 بطلب من كلية التكنولوجيا والعلوم التطبيقية لإثراء المقرر بالمادة العملية ومساعدتك على فهم التطبيق العملي لهذا المقرر. قسم هذا الدليل إلى خمسة أقسام. القسم الأول ناقش الأساسيات اللازمة لتجهيز الجهاز لتنفيذ أوامر الرسم . والقسم الثاني ناقش الدوال الجاهزة لرسم الأشكال الهندسية وتلوينها وتحريكها . القسم الثالث ناقش كتابة برامج كاملة باستخدام دوال الرسم الجاهزة وتلوينها وتحريكها. القسم الرابع ناقش كتابة برامج كاملة لرسم الأشكال الهندسية باستخدام خوارزميات خاصة، وبدون استخدام الدوال الجاهزة بلغة سي. القسم الخامس ناقش استخدام الماوس على شاشة الرسم. هذا الدليل صمم إلى طلبة مقرر الرسم بالحاسوب ولمن يرغب تعلم برمجة الأشكال الهندسية بلغة سي وكتابة النصوص بشكل رسومي، من الممكن استخدام برمجة الأشكال والرسومات لتطوير الألعاب الخاصة.

القسم الأول:

في هذا القسم سوف يتم مناقشة أساسيات الرسم بلغة سي وتعريف الدوال الأساسية اللازمة لهذا الهدف:

عند برمجة الرسومات بلغة سي عليك استخدام مكتبة الدوال المعيارية لإنجاز عملك (أهدافك)، وذلك بتمرير معاملاتك إلى الدالة وهذا كل ما تريد عمله , في هذا الدليل نجد تقريباً كل الدوال مع الشرح الموجز ومثال برمجي ليبين استخدام الدالة الرسومية. عادة الشاشة التي يتم استخدامها في DOS شاشة النمط النصي. وللرسم أنت بحاجة إلى تعيين نمط رسومي للشاشة. وحتى يحدث ذلك أنت بحاجة لتضمين ملف الترويسة graphics.h. في برنامج لغة سي أولاً أنت بحاجة لتعيين قيم أولية إلى مشغلات الرسم على جهاز الحاسوب، ويتم ذلك باستخدام دالة initgraph() الموجودة في مكتبة graphics.h. سوف نتعرف على الدالة initgraph() والتي تستخدم لتعيين قيم أولية إلى كرافيك مود graphic mode initialize. ودالة initgraph() موجودة في ملف الترويسة graphics.h وتستخدم لتنشيط الوضع الرسومي على الحاسب ولهذا كل برنامج رسومي يجب أن يتضمنها ويتضمن ملف الترويسة "graphics.h".

1.1 دالة initgraph()

تستخدم هذه الدالة لتعيين قيم أولية لوضع (نمط) الشاشة المناسب للرسم graphic mode وتمسح الشاشة حيث يوجد وضعين (نمطين) للشاشة وهي Text Mode و Graphics Mode. تستخدم للتحويل من الشاشة العادية النصية إلى الشاشة الرسومية.

الصيغة العامة لهذه الدالة هي:

```
void initgraph(int far *graphdriver, int far *graphmode, char far *pathtodriver);
```

معاملات هذه الدالة هي عنوان gd وعنوان gm ومسار مكان وجود ملفات الرسم (BGI)على القرص الصلب في جهازك. لبدء نظام الرسم يجب استدعاء الدالة initgraph حيث تعمل هذه الدالة على تعيين قيم أولية لنظام الرسم مع ملف مكتبة graphics.h الرسومات وتغييرها إلى شاشة الرسم وذلك بتحميل مشغل الرسم graphic drive من القرص ووضع النظام في وضع (نمط) الرسم graphic mode وملف BGI في مجلد البرنامج. وتعمل على إعادة كل إعدادات الرسم لوضعها الافتراضي (اللون، المكان الحالي، palette، viewport) وإعادة وضع graphresult إلى صفر (0). وهي الخطوة الأولى التي نحتاجها من أجل كتابة برامج الرسم . بحاجة لتمرير المتغيرات gd,gm للدالة وعادة تستخدم القيم الآتية: gd=VGA و gm=VGAHI للرسم وإحداثيات هذا النوع من الشاشات هو 640x480 pixels. ويمكن استخدام الدالة detectpraph() لإيجاد قيم gd, gm. بعد ذلك يصبح الجهاز قادراً على استقبال أوامر الرسم المختلفة.

2.1 دالة detectgraph()

driver عدد صحيح يحدد مشغل الرسم المراد ا ستخدامه، ويمكن إسناد قيمة لمشغل الرسم باستخدام ثابت من الأنواع المتعددة لمشاغل الرسم المعرفة في الملف graphics.h. وهي موضحة في القائمة التالية:
Graphicmode عدد صحيح يحدد القيمة الأولية ل graphics mode إلا إذا استخدم الصيغة graphdriver=DETECT* في هذه الحالة initgraph تضع graphmode* لأعلى دقة وضوح متوفرة للمشغل الذي تم الكشف (البحث) عنه. ويمكن إسناد قيمة إلى graphmode* تستخدم للكشف عن قيم مشغل الرسم الحالي graphic driver وحالة الشاشة graphic mode وناتج هذه الدالة يستخدم في الدالة initgraph() كمعاملات إدخال بالإشارة. Detectgraph تكشف graphics adapter لنظامك إذا لم يكشف عن إي وحدة رسم مادية فإن graphdriver* يتم تعيينه إلى grNotDetected(-2) و graphresult تعيد grNotDetected(-2).

```
detectgraph(&grd,&grm);
```

Graphicباستخدام ثابت من الأنواع المتعددة ل graphics_modes. في graphics mode كل إحداثيات الشاشة تستخدم البكسل كوحدة قياس. وعدد البكسلات على الشاشة يحدد دقة وضوح الشاشة. كما هو موضح في القائمة التالية:
Graphdriver and graphmode يجب تعينها بقيمة صحيحة من الجداول التالية وإلا سوف تحصل على قيم غير متوقعة. الإستثناء هو graphdriver=DETECT

graphics_drivers constant	Numeric value
DETECT	0 (requests autodetect)
CGA	1
MCGA	2
EGA	3
EGA64	4
EGAMONO	5
IBM8514	6
HERCMONO	7
ATT400	8
VGA	9
PC3270	10

Pathtodriver تحدد مسار المجلد أين initgraph تبحث أولاً عن graphics drivers(*.BGI) إذا لم تجده initgraph تبحث في المجلد الحالي . وإذا كان المسار إلى المشغل فارغ فإن ملفات المشغل يجب أن تكون في المجلد الحالي. إذا كان ملف BGI في نفس مجلد برنامجك نترك المسار فارغاً ""

graphics	GRAPHICS MODE	Columns		
Driver	graphics_mode	Value	x Rows	Palette
CGA	CGAC0	0	320 x 200	C0
	CGAC1	1	320 x 200	C1
	CGAC2	2	320 x 200	C2
	CGAC3	3	320 x 200	C3
	CGAHI	4	640 x 200	2 color
EGA	EGALO	0	640 x 200	16 color
	EGAHI	1	640 x 350	16 color
EGA-MONO	EGAMONOHI	3	640 x 350	2 color
	EGAMONOHI	3	640 x 350	2 color
VGA	VGALO	0	640x 200	16 color
	VGAMED	1	640x 350	16 color
	VGAHI	2	640x 480	16 color

3.1 دالة getgraphmode()

تستخدم هذه الدالة للحصول على وضع الرسم الحالي والقيمة المعادة تكون عدد صحيح.
الصيغة العامة لهذه الدالة هي:

```
int getgraphmode(void);
```

مثال:
لو أردنا معرفة وضع (نمط) الشاشة الحالي للجهاز
الحل:
نكتب الجمل التالية:

```
// getgraphmode returns the current graphics mode
    int gm;
    gm = getgraphmode();
    printf("current mode: %d", gm);
```

4.1 دالة setgraphmode()

تستخدم هذه الدالة لتعيين أو تغيير وضع (نمط) الشاشة إلى شاشة الرسم ومسح الشاشة.

مثال:
لو أردنا تعيين قيمة للنظام ليصبح في وضع الرسم
الحل:
نكتب أي من الجمل التالية:

```
setgraphmode(VGAMED);
setgraphmode(VGAHI);
```

5.1 دالة getdrivename()

تستخدم هذه الدالة لإعادة اسم المشغل الذي تم تحميله حالياً ، أي تعيد مؤشر إلى مشغل الرسم الحالي.
الصيغة العامة لهذه الدالة:

```
char* getdrivename(void);
```

مثال:
لو أردنا معرفة نوع مشغل الرسم الحالي للجهاز.
الحل:
نكتب الجمل التالية:

```
 char *drivername;
initgraph(&gd, &gm, "C:\\TC\\BGI");
drivername = getdrivername();
outtextxy(200, 200, drivername);
```

6.1 دالة grapherrormsg()

تستخدم هذه الدالة لتعيد نص رسالة الخطأ
الصيغة العامة لهذه الدالة هي:

```
char *grapherrormsg( int errorcode );
```

مثال:
لو أردنا طباعة نص رسالة الخطأ
الحل:
نكتب الجمل التالية:

```
 int gd, gm, errorcode;
detectgraph(&grd,&grm);
 initgraph(&gd, &gm, "C:\\TC\\BGI");
   errorcode = graphresult();
    if(errorcode != grOk)  {
```

```
    printf("Graphics error: %s\n", grapherrormsg(errorcode));
    exit(1);  }
```

7.1 دالة graphresult()

تستخدم هذه الدالة للحصول على رمز الخطأ errorcode بعد استدعاء الدالة initgraph() والدالة grapherrormsg. دالة graphresult تعيد رمز الخطأerrorcode عن أي عملية رسم يحدث عنها خطأ. وتستقل الدالة grapherrormsg رمز الخطأ. إذا لم يحدث أي خطأ عن عمليات الرسم سوف تعيد grOk.
لكي نحصل على خطأ احذف ملف BGI.* من المجلد الحالي أو مجلد BGI من مسار الإعدادات، سوف تحصل على رسالة خطأ device drive file not found (EGAVGA.BGI)

```
int errorcode = graphresult();
   if(errorcode != grOk)  {
          printf(grapherrormsg(errorcode);
        getch();
        return 0;   }
```

القيمة المعادة:
دائما دالة initgraph() تضع قيمة error code في وضع النجاح للعملية وهي الصفر (0) وفي حالة حدوث خطأ (فشل العملية) يتم وضع تعيين القيم -2, -3, -4, or -5 ل graphic driver. و graphresult تعيد نفس القيم الموضحة في الجدول التالي:

Constant Name	Number	Meaning
grNotDetected	-2	Cannot detect a graphics card
grFileNotFound	-3	Cannot find driver file
grInvalidDriver	-4	Invalid driver
grNoLoadMem	-5	Insufficient memory to load driver

8.1 دالة closegraph()

تستخدم هذه الدالة لإغلاق شاشة (نافذة) الرسم graphics mode والعودة إلى شاشة الكتابة العادية (الشاشة النصية) text mode أو الشاشة التي كانت عليها قبل استدعاء شاشة الرسم. وتمسح الشاشة أيضاً. يجب وضعها في نهاية برنامج الرسم. هذه الدالة تلغي حجز الذاكرة الذي تم حجزه من قبل نظام الرسم، واسترجاع الشاشة إلى وضعها الذي كانت عليه قبل استدعاء دالة initgraph().
إذا أردنا استخدام كل من الشاشة النصية وشاشة الرسم في برنامج، يجب استخدام الدوال initgraph() و closegraph() في البرنامج.
الصيغة العامة لهذه الدالة هي:

```
void closegraph();
```

مثال:
لو أردنا التحويل من الشاشة النصية إلى شاشة الرسم وكتابة العبارة التالية على شاشة الرسم ثم إغلاق شاشة الرسم والعودة إلى الشاشة النصية. Press any key to close the graphics mode. .

الحل:
نكتب الجمل الآتية:

```
initgraph(&gd, &gm, "C:\\TC\\BGI");
outtext("Press any key to close the graphics mode...");
getch();
closegraph();
```

تدريب 1:
ما هي وظيفة البرنامج التالي:

```
#include<stdio.h>
```

```
#include<conio.h>
#include<graphics.h>
void main()
{
   int gd=DETECT,gm;
  initgraph(&gd,&gm,"c:\\tc\\bgi"□□);
  outtext("□□Enter any key to come outside from the graphics mode"□□);
   getch();
 closegraph();
printf("Text Mode ");
}
```

الحل:

الدالة initgraph() استخدمناها للتحول من الشاشة النصية text mode إلى شاشة الرسم graphics mode.

الدالة outtext() استخدمناها لكتابة النص على شاشة الرسم عند النقطة التي إحداثياتها (0 ,0).

الدالة closegraph() استخدمناها لإغلاق شاشة الرسم والإنتقال إلى الشاشة النصية.

الدالة getch() استخدمناها للإنتظار حتى الضغط على أي زر من أزرار لوحة المفاتيح.

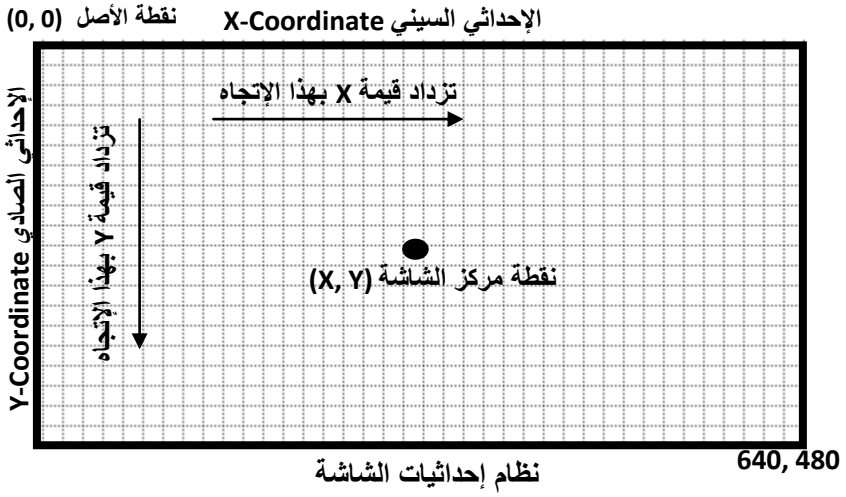
الدالة printf() استخدمناها لكتابة النص على الشاشة العادية (النصية).

القسم الثاني:

بعد هذه المقدمة والتعرف على الدالة initgraph وغيرها من الدوال يمكننا الآن استخدام دوال الرسم لرسم الأشكال مثل النقطة والدائرة والمستطيل والخطوط.....الخ. وكذلك تعلم كيف تغير الألوان والخطوط باستخدام الدو ال المناسبة وكتابة النصوص بأنواع خطوط مختلفة وتغيير الحجم واللون ....الخ. وبعد ذلك يمكنك الاستمرار لاستخدام دوال الصور مثل getimage, putimage لعمل الحركة. وأخيراً يمكننا عمل برامج رسومية وحركات ومشاريع وألعاب . فيما يلي جدول بدوال الرسم الجاهزة بلغة سي والتي سنوضح معظمها في هذا القسم .

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

وقبل القيام بعملية الرسم يجب معرفة نظام الإحداثيات الذي يمثل موقع الرسم بالنسبة للأشكال. حيث يتكون هذا النظام من الإحداثي السيني (x-coordinates) الذي يمثل البعد الأفقي horizontal للشكل. والإحداثي الصادي -y) (coordinates الذي يمثل البعد العمودي vertical للشكل. تبدأ الإحداثيات بالنقطة (0, 0) الموجودة في أعلى يسار الشكل وتدعى نقطة الأصل، وعند زيادة قيمة الإحداثي السيني وثبات قيمة الإحداثي الصادي يتم التحرك أفقياً باتجاه اليمين، وعند زيادة قيمة الإحداثي الصادي وثبات قيمة الإحداثي السيني يتم التحرك عمودياً إلى الأسفل، وعند زيادة قيمة الإحداثي السيني والصادي معاً يتم التحرك باتجاه مركز (منتصف) الشكل.



البرنامج التالي يبين الهيكل العام لكل البرامج الرسومية بلغة سي:

```
#include<graphics.h>
#include<conio.h>
int main(){
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  getch();
  closegraph();
  return 0;  }
```

هذا المقطع البرمجي يعين القيم الأولية إلى graphic mode و graphic drive.
المتغيرات gm, gd ترمز هذه الإختصارات إلى graphic mode وكذلك graphic driver ويمكن استخدام إي أسماء أخرى.
كلمة DETECT عبارة عن ماكرو معرف في ملف الترويسة "graphics.h" يستخدم للكشف عن جهاز الرسم المناسب (التعرف على ميزات كارت الرسم الموجود).
الدالة ()initgraph أوتوماتيكياً تقرر مشغل الرسم graphic driver المناسب، و وضع (نمط) الشاشة المناسب graphic mode وهي القيمة العظمى لدقة وضوح الشاشة.
الدالة ()getch تساعد على الانتظار بعض الوقت (تثبيت الصورة على الشاشة) لحين الضغط على أي زر.
الدالة ()closegraph تغلق graphics mode أي تحول من وضع الرسم إلى وضع الكتابة العادية (الشاشة النصية).
include<graphics.h> تستخدم لاستدعاء (تضمين) مكتبة الرسم في البرنامج.

بعد هذه المقدمة البسيطة والسريعة لهيكل برنامج الرسم حان الوقت لنتكلم عن دوال الرسم بشكل أكثر تفصيلاً.

1.2 دالة ()cleardevice

تستخدم هذه الدالة لمسح شاشة الرسم من الرسومات والنص المكتوب عليها ولون الخلفية ووضع المشيرة على نقطة الأصل (0, 0) في الزاوية العلوية اليسرى من الشاشة وتعبئة الشاشة بلون الخلفية الحالي. هذه الدالة ليس لها معاملات.
الصيغة العامة لهذه الدالة هي:

```
void cleardevice();
```

مثال:
لو أردنا استخدام الدالة cleardevice في برنامج لمسح شاشة الرسم.
الحل:
نكتب الجمل الآتية:

```
outtext("Press any key to clear the screen.");
getch();
cleardevice();
outtext("Press any key to exit...");
```

ملاحظة:
لا تستخدم دالة مسح الشاشة النصية clrscr() لمسح شاشة الرسم.

2.2 دالة putpixel()

تستخدم هذه الدالة لرسم نقطة ملونة في مكان ما على الشاشة وهذا المكان يمثل بإحداثيات (X, Y)، حيث أن مقدار الإزاحة على محور السينات يمثل بالرمز X ومقدار الإزاحة على محور الصادات ويمثل بالرمز Y.
الصيغة العامة لهذه الدالة:

```
void putpixel(int x, int y, int color);
```

معاملات هذه الدالة 3 متغيرات من النوع العددي الصحيح. المتغير الأول X والمتغير الثاني Y وتمثل إحداثيات النقطة، والمتغير الثالث Color يمثل رقم لون النقطة.

مثال:
لو أردنا رسم نقطة خضراء في المكان الذي إحداثياته السينية والصادية هي (35, 45) حيث أن x=35, y=45.
الحل:
نكتب الجملة الآتية:

```
putpixel(35, 45, GREEN);
```

أو

```
putpixel(x, y, 2);
```

ملاحظة:
أسماء الألوان تكتب بأحرف كبيرة Capital letters. أو يكتب رقم اللون حيث أن الألوان مرتبة من صفر (0) إلى (15). اما باقي الدوال فلا يوجد بها متغير اللون لذا نستخدم معهم دالة تغيير اللون.
إحداثيات النقطة يمكن أن تكون أعداد صحيحة أو كسور عشرية.

ويمكن استخدام الدالة putpixel أيضاً لرسم الدوائر والخطوط المستقيمة والأشكال البيضاوية باستخدام خوارزميات مختلفة.

تدريب 2:
اكتب برنامجاً بلغة سي لرسم نقطة حمراء إحداثياتها السينية والصادية هي (320, 240). باستخدام الدالة putpixel.

الحل:

```
#include<graphics.h>
#include<conio.h>
int main()
{
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  putpixel(320, 240, RED);
  getch();
  closegraph();
  return 0;   }
```

تدريب 3:
اكتب برنامجاً بلغة سي لرسم خط مستقيم أفقي بلون أزرق باستخدام الدالة putpixel، من النقطة (10, 10) إلى النقطة (200, 10).

الحل:
إن حل هذا السؤال يتطلب تكرار تنفيذ الدالة putpixel ولهذا نلجأ إلى استخدام حلقات التكرار (الدوران) مثلاً حلقة for على النحو الآتي:.

```
#include<graphics.h>
#include<conio.h>
void main()
{
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  for(int i=10; i<= 200; i++)
    putpixel(i, 10, BLUE);
  getch();
  closegraph();
}
```

3.2 دالة getpixel()

تستخدم هذه الدالة للحصول على (إعادة) رقم لون النقطة الحالي (pixel) في أي موقع على الشاشة إحداثياته (x, y) .
الصيغة العامة لهذه الدالة هي:

```
int getpixel (int x, int y);
```

معاملات هذه الدالة هي إحداثيات النقطة على محور السينات وعلى محور الصادات.
مثال:
لو أردنا معرفة لون نقطة على الشاشة إحداثياتها (0, 0).
الحل:
نكتب الجمل التالية:

```
char array[50];
color = getpixel(0, 0);
sprintf(array,"color of pixel at (0,0) = %d",color);
outtext(array);
```

ناتج هذه الجمل يكون الصفر (0) أي رقم اللون الأسود، وهو اللون الإفتراضي لخلفية الشاشة.

تدريب4:
اكتب برنامجاً بلغة سي لرسم نقطة حمراء على الشاشة إحداثياتها (320, 240). ثم طبلعة رقم لون النقطة نفسها على الشاشة مع رسالة توضح ذلك.

الحل:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
void main()
 {
        int gd=DETECT, gm, color;
        char a[50];
        initgraph(&gd, &gm,"c:\\tc\\bgi");
        putpixel(320, 240, RED);
```

```
        color=getpixel(320,240);
        sprintf(a,"color of pixel at location (320,240) is %d",color);
        outtextxy(30, 100, a);
         getch();
         closegraph();
}
```

4.2 دالة line()

تستخدم هذه الدالة لرسم الخطوط المستقيمة من مكان نقطة البداية (x_0, y_0) إلى نقطة آخرى على الشاشة نقطة النهاية (x_1, y_1) بمعنى أن هذه النقاط تمثل نهايات الخط.

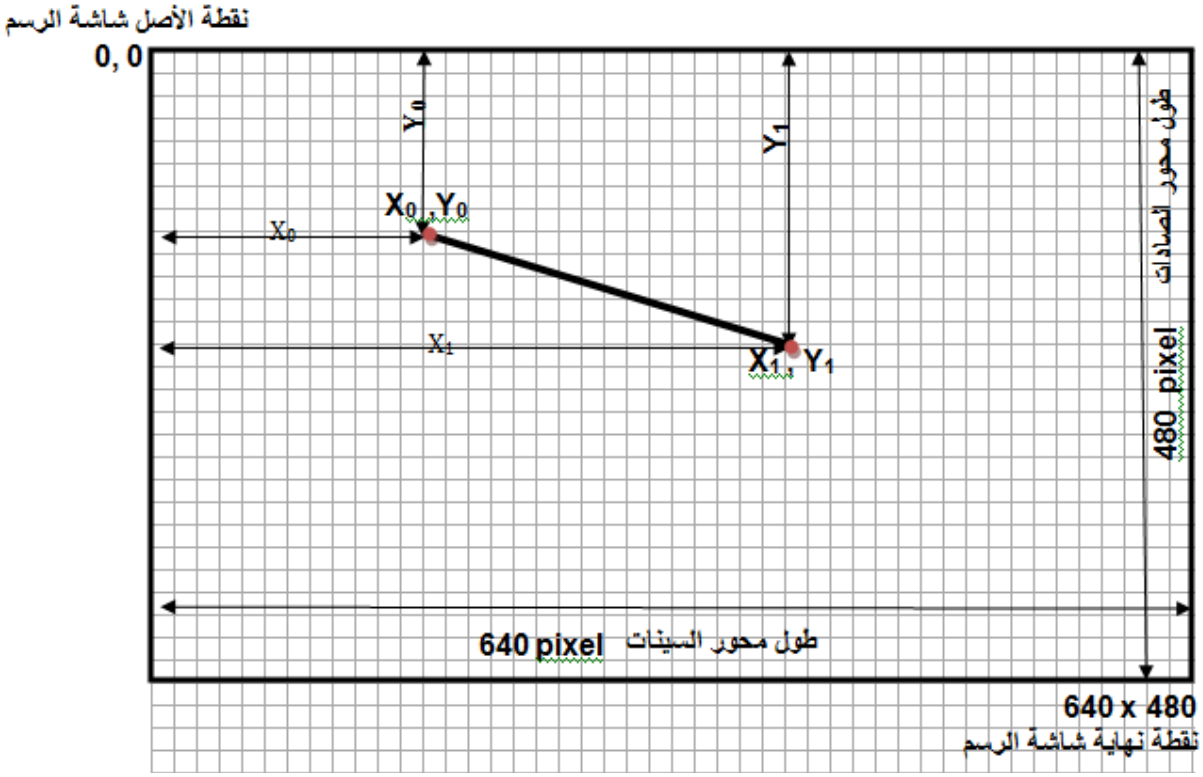
الصيغة العامة لهذه الدالة هي:

void line(int x_0, int y_0, int x_1, int y_1);

معاملات هذه الدالة هي 4 متغيرات عددية تمثل إحداثيات نقطة بداية الخط (x_0,y_0) وإحداثيات نقطة نهاية الخط (x_1, y_1).

ملاحظة: معاملات هذه الدالة يمكن أن تكون متغيرات عددية صحيحة أو كسور عشرية.

الشكل التالي يوضح إحداثيات شاشة الرسم التي تبدأ من نقطة الأصل (0, 0) إلى نقطة النهاية (640, 480) وإحداثيات خط مستقيم حيث أن X_0 تمثل بعد نقطة بداية الخط المستقيم من نقطة الأصل على محور السينات، Y_0 تمثل بعد نقطة بداية الخط المستقيم من نقطة الأصل على محور الصادات. X_1 تمثل بعد نقطة نهاية الخط المستقيم من نقطة الأصل على محور السينات، Y_1 تمثل بعد نقطة نهاية الخط المستقيم من نقطة الأصل على محور الصادات.



مثال:

لو أردنا أن نرسم خطاً من النقطة (10, 10) إلى النقطة (120, 120).

الحل:

نكتب الأمر الآتي: `line(10, 10, 120,120);`

ويمكن استخدام هذه الدالة لرسم الأشكال متعددة الأضلاع مثل المثلث والمربع والمستطيل والشكل الخماسي ...الخ. باستخدام الدالة line عدة مرات وبنقاط بداية ونهاية مختلفة.

تدريب 5:

اكتب برنامجاً بلغة سي لرسم خطاً مستقيماً من نقطة البداية (100, 100) إلى نقطة النهاية (200, 100).

الحل:

```
#include <graphics.h>
#include <conio.h>
main()
{  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  line(100, 100, 200, 100);
   getch();
  closegraph();  }
```

تدريب 6:
اكتب برنامجاً بلغة سي لرسم مثلث باستخدام الدالة line عدة مرات، إحداثيات نقاط رؤوس المثلث هي: A(100, 200), B(300, 200), C(200, 100)

الحل:

```
#include <graphics.h>
#include <conio.h>
main()
{  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  line(100, 200, 300, 200);
  line(300, 200, 200, 100);
  line(200, 100, 100, 200);
  getch();
  closegraph();
  return 0;    }
```

ملاحظة:
يمكن رسم الخطوط بألوان مختلفة باستخدام دالة تغيير الألوان()setcolor. ويجب أن تكتب قبل دالة الرسم

5.2 دالة setcolor()

تستخدم هذه الدالة لتحديد اللون الأمامي (لون خط الرسم) على شاشة الرسم ولتغيير لون خط الرسم الحالي. لغة سي توفر العديد من الألوان وأن كل لون متوفر ويقابله رقم، عدد الألوان يعتمد على نوع graphics mode and driver الحالي. ويمكن معرفة مجموع عدد الألوان المختلفة والمتوفرة باستخدام الدالة getmaxcolor(). أرقام الألوان الممكنة من 0 إلى 15 بالترتيب الآتي بمجمل 16 لوناً:

BLACK(0), BLUE(1), GREEN(2), CYAN(3), RED(4), MAGENTA(5), BROWN(6), LIGHTGRAY(7), DARKGRAY(8), LIGHTBLUE(9), LIGHTGREEN(10), LIGHTCYAN(11), LIGHTRED(12), LIGHTMAGENTA(13), YELLOW(14), WHITE(15) and BLINK (128).

الصيغة العامة لهذه الدالة:

```
void setcolor(int color);
```

ملاحظة:
يمكن كتابة اسم اللون بين الأقواس بأحرف كبيرة بدلاً من رقم اللون كما يلي: setcolor(RED);
اللون الافتراضي لخط الرسم إذا لم يحدد هو اللون الأبيض.

مثال:
إذا أردنا رسم خط مستقيم بلون أحمر نقاط نهاياته هي: (100, 100), (200,100)
الحل:
نكتب الجمل التالية:

1. نحدد لون خط الرسم باستخدام الدالة setcolor() كما يلي: setcolor(4); او setcolor(RED)
2. نكتب أمر رسم الخط المستقيم كما يلي: line(100, 100, 200, 100);

تدريب7:
اكتب برنامجاً بلغة سي لرسم مربع ألوان أضلاعه هي الأبيض، والأحمر، والأزرق، الأصفر

الحل:

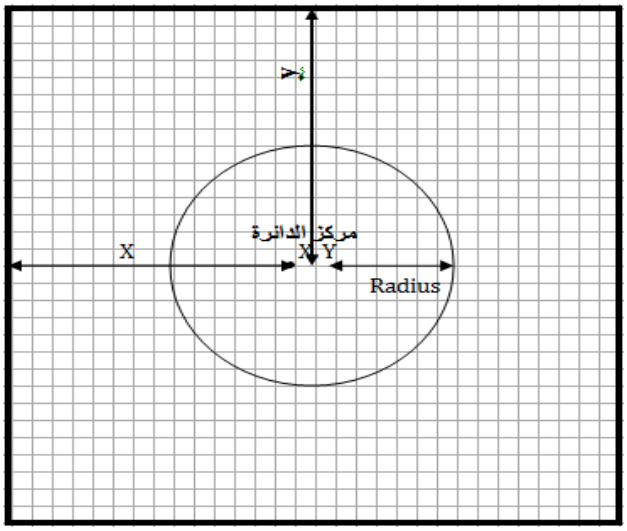
```
#include<graphics.h>
#include<conio.h>
main()
{  int gd = DETECT, gm;
  initgraph(&gd,&gm,"C:\\TC\\BGI");
  line(100,100,100,200);        /* drawn in white color */
  setcolor(RED);
  line(100,200,200, 200);        /* drawn in red color   */
  setcolor(1);
  line(200, 200, 200, 100);      /* drawn in blue color */
  setcolor(14);
  line(200, 100, 100, 100);      /* drawn in yellow color */
  getch();
  closegraph();
  return 0;    }
```

6.2 دالة circle()

تستخدم هذه الدالة لرسم دائرة بعد تحديد نقطة مركز رسم الدائرة وطول نصف قطرها.
الصيغة العامة لهذه الدالة هي:

```
void circle(int x, int y, int radius);
```

معاملات هذه الدالة هي المتغيرات العددية (X, Y) وتمثل نقطة مركز الدائرة والمتغير العددي Radius يمثل نصف قطر الدائرة.
الشكل التالي يبين نقطة مركز الدائرة ونصف قطرالدائرة وبعد نقطة المركز عن نقطة الأصل على محور السينات وبعد نقطة المركز عن نقطة الأصل على محور الصادات.



مثال:
إذا أردنا أن نرسم دائرة طول نصف قطرها 100 ومركزها نقطة منتصف الشاشة (320, 240) بلون أبيض وخلفية سوداء
الحل:
نكتب الأمر الآتي: circle(320, 240, 100);

تدريب 8:
اكتب برنامجاً بلغة سي لرسم دائرة بلون أبيض على خلفية صفراء احداثيات نقطة مركزها (150, 150) وطول نصف قطرها 70.

الحل:

```
#include<graphics.h>
#include<conio.h>
 main()   {
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");

  setbkcolor(14);
  setcolor(15);
  circle(150, 150, 70);
   getch();
  closegraph();
  return 0;   }
```

7.2 دالة getcolor()

تستخدم هذه الدالة للحصول على (إعادة) رقم لون خط الرسم الحالي ولون خط الرسم الإفتراضي هو اللون الأبيض ورقمه (15).
الصيغة العامة لهذه الدالة هي:

```
int getcolor();
```

مثال:
لو أردنا معرفة لون خط الرسم الحالي وطباعة رقمه مع رسالة توضح ذلك
الحل:
نكتب الجمل التالية:

```
char a[100];
int drawing_color = getcolor();
sprintf(a,"Current drawing color = %d", drawing_color);
outtextxy( 10, 10, a );
```

8.2 دالة setbkcolor()

تستخدم هذه الدالة لتعيين (تغيير) لون خلفية الرسم الحالية في graphics mode وينطبق على هذه الدالة كل ما ينطبق على الدالة setcolor() من حيث عدد الألوان وأرقامها ومعرفة أقصى لون. وأن لون خلفية الرسم الافتراضي هو اللون الأسود. إذا لم يحدد اللون يعتمد اللون الافتراضي ورقمه الصفر (0). ويجب أن تكتب قبل دالة الرسم.

الصيغة العامة لهذه الدالة هي:

```
void setbkcolor(int color);
```

مثال:
لو أردنا وضع لون خلفية الرسم أبيض ورسم خط بلون أزرق.

الحل:
نكتب الجمل التالية:

```
setbkcolor(15);or setbkcolor(WHITE);
setcolor(1); or setcolor(BLUE);
line(200, 200, 200, 100);
```

تدريب 9:
اكتب برنامجاً بلغة سي يرسم دائرة خضراء نصف قطرها 100 وإحداثيات مركزها (320, 240)، ثم اكتب فوقها العبارة "It is Circle" بلون أحمر عند الإحداثيات (300,120). ثم اطبع رقم لون خط الرسم الحالي مع رسالة توضح ذلك أسفل الدائرة عند الإحداثيات (280,380).

الحل:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
 void main()
    {          int gd=DETECT, gm, color;
         char a[50];
          initgraph(&gd,&gm,"c:\\tc\\bgi");
               setcolor(GREEN);
         circle(320,240,100);
         setcolor(RED);
         outtextxy(300,120."It is circle");
          color=getcolor();
         sprintf(a,"The value of the current drawing color is %d",color);
         outtextxy(280, 380, a);
       getch();
       closegraph();      }
```

تدريب 10:
اكتب برنامجا بلفة سي يرسم دائرة بيضاء نصف قطرها 100 عند نقطة منتصف الشاشة، على خلفية خضراء.

الحل:

```
#include<graphics.h>
#include<conio.h>
main()
{
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  setbkcolor(GREEN);
  circle(320, 240, 100);
  getch();
  closegraph();
  return 0;   }
```

9.2 دالة getbkcolor()

تستخدم هذه الدالة للحصول على (إعادة) رقم لون خلفية شاشة الرسم الحالية. علماً بأن لون خلفية شاشة الرسم الإفتراضي هو الأسود ورقمه صفر (0).
الصيغة العامة لهذه الدالة هي:

```
int getbkcolor();
```

مثال:
لو أردنا طباعة لون خلفية شاشة الرسم مع رسالة توضح ذلك على شاشة الرسم عند النقطة (10, 10).
الحل:
نكتب الجمل التالية:

```
char a[100];
```

```
int bkcolor = getbkcolor();
sprintf(a,"Current background color = %d", bkcolor);
outtextxy( 10, 10, a);
```

تدريب 11:
اكتب برنامجاً يرسم دائرة بيضاء على خلفية خضراء نصف قطرها 100 ونقطة مركزها منتصف الشاشة وهي (320, 240). ثم اطبع لون خلفية شاشة الرسم مع رسالة توضح ذلك.

الحل:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
 void main()
 {        int gd=DETECT, gm,bkcolor;
         char a[50];
           initgraph(&gd,&gm,"c:\\tc\\bgi");
           setbkcolor(GREEN);
           circle(320,240,100);
           bkcolor=getbkcolor();
           sprintf(a,"The value of the current background color is %d",bkcolor);
           outtextxy(100,240,a);
           getch();
           closegraph();  }
```

10.2 دالة getmaxcolor()

تستخدم هذه الدالة للحصول على أو إعادة قيمة عددية لآخر لون في سلسلة الألوان لمشغل الرسم وحالة شاشة الرسم الحالية ويكون الرقم (15) عادةً. ولمعرفة الألوان وأرقامها الموجودة بلغة سي تتبع القائمة أعلاه التي تحتوي على رقم اللون واسمه. عدد الألوان الموجودة لمشغل الرسم ونمطه يعبر عنها بالصيغة التالية: getmaxcolor()+1 لأن الترقيم يبدأ من الصفر(0) وينتهي بالرقم 15.

الصيغة العامة لهذه الدالة هي:

```
int getmaxcolor();
```

مثال:
لو أردنا كتابة النص الآتي Maximum number of colors for current graphics mode and driver على شاشة الرسم بلون تحصل عليه من استخدام الدالة getmaxcolor() .

الحل:

نكتب الجمل التالية:

```
setcolor(getmaxcolor());
outtext("Maximum number of colors for current graphics mode and driver";
```

تدريب 12:
اكتب برنامجاً بلغة سي لطباعة النص الآتي "Alquds is the city of peace" بلون احصل عليه من الدالة getmaxcolor() على شاشة الرسم بالموقع الذي إحداثياته السينية 100 والصادية 100. ثم اطبع رقم اللون مع رسالة توضيحية عند النقطة 100, 240.

الحل:

```
#include<graphics.h>
#include<conio.h>
 main()
```

```
{  int gd = DETECT, gm, maxcolor;
   char a[50];
   initgraph(&gd,&gm,"C:\\TC\\BGI");
   maxcolor = getmaxcolor();
   setcolor(maxcolor);
   outtextxy(100, 100, "Alquds is the city of peace");
   sprintf(a,"The maximum color value is:%d",maxcolor);
   outtextxy(100, 240,a);
   getch();
   closegraph();
   return 0;  }
```

11.2 دالة rectangle()

تستخدم هذه الدالة لرسم المستطيلات او المربعات محددة أطوال الأضلاع والممثلة بإحداثيات الزاوية العلوية اليسرى والزاوية السفلية اليمنى.
الصيغة العامة لهذه الدالة هي:

```
void rectangle(int left, int top, int right, int bottom);
```

معاملات هذه الدالة هي أرقام إحداثيات رسم المستطيل من الزاوية العلوية اليسرى إلى الزاوية السفلية اليمنى. حيث أن
المتغير left يمثل إحداثيات X للزاوية العلوية اليسرى. المتغير top يمثل إحداثيات Y للزاوية العلوية اليسرى
المتغير right يمثل إحداثيات X للزاوية السفلية اليمنى. المتغير bottom يمثل إحداثيات Y للزاوية السفلية اليمنى

ملاحظة:
إذا كان طول الضلع على المحور السيني يساوي طول الضلع على المحور الصادي نحصل على شكل مربع.
ويمكن تحديد لون خط رسم المستطيل باستخدام الدالة setcolor() وتحديد لون خلفية الرسم باستخدام الدالة setbkcolor() ويمكن تعبأة الأشكال المغلقة مثل المستطيل والمربع والدائرة بلون باستخدام دالة floodfill().
مثال:
لو أردنا رسم مستطيل أبيض إحداثياته العلوية اليسرى (10, 10) وإحداثياته السفلية اليمنى (300,200). على خلفية سوداء.
الحل:
نكتب الأمر الآتي: `rectangle(10, 10, 300,200);`

تدريب 13:
اكتب برنامجاً بلغة سي لرسم مستطيل لونه أصفر على خلفية حمراء إحداثيات الزاوية العلوية اليسرى له (100,100) وإحداثيات الزاوية اليمنى السفلية (400,250).

الحل:

```
#include<graphics.h>
#include<conio.h>
 main()
{  int gd = DETECT, gm;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
    setbkcolor(RED);
   setcolor(YELLOW);
   rectangle(100,100,400,250);
   getch();
   closegraph();
   return 0;   }
```

12.2 دالة floodfill()

تستخدم هذه الدالة لتعبئة الأشكال المغلقة بلون ونمط التعبئة الحالي (الإفتراضي) وهو اللون الأبيض. وتكتب بعد دالة الرسم.

الصيغة العامة لهذه الدالة هي:

```
void floodfill(int x, int y, int border);
```

معاملات هذه الدالة هي المتغيرات (x, y) وتمثل إحداثيات نقطة داخل المنطقة المراد تعبأتها والمتغير border لون خط رسم المنطقة المراد صب اللون بداخله. إذا لم يحدد لون التعبئة يعبأ الشكل بلون أبيض وهو اللون الافتراضي. وإذا كانت النقطة خارج الشكل يتم تعبئة المنطقة المحيطة بالشكل. ويمكن استخدام الدالة setfillstyle لتغيير لون وشكل نمط التعبئة.

مثال:

لو أردنا رسم دائرة نصف قطرها 60 ومركزها 100و100 وتعبأتها باللون الإفتراضي.

الحل:

نكتب الجمل التالية:

```
circle(100, 100, 60);
floodfill(101,101, 15);
```

لون border يجب أن يكون نفس اللون الإفتراضي.

مثال:

لو أردنا رسم الدائرة السابقة بلون أحمر وتعبأتها بلون أبيض، اللون الإفتراضي.

الحل:

نكتب الجمل التالية:

```
setcolor(RED);
circle(100, 100, 60);
floodfill(101, 101, 4);
```

تدريب 14:

اكتب برنامجاً بلغة سي لرسم مستطيل بلون أصفر ومعبأ بلون أبيض اللون الإفتراضي أحداثيات الزاوية العلوية اليسرى (100, 100) وإحداثيات الزاويا السفلية اليمنى (300, 200).

الحل:

```
#include <graphics.h>
#include <conio.h>
 main()
{   int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");
   setcolor(14);
   rectangle(100,100,300,200);
   floodfill(100,100,YELLOW);
   getch();
   closegraph();
   return 0; }
```

13.2 دالة setfillstyle()

تستخدم هذه الدالة لتحديد لون وشكل نمط التعبئة. وترسم الأشكال بعدها مباشرة. وتستخدم مع دالة تلوين الأشكال الهندسية floodfill.

الصيغة العامة لهذه الدالة هي.

```
void setfillstyle(int pattern, int color);
```

معاملات هذه الدالة هي المتغير pattern ويحدد شكل نمط التعبئة والمتغير color يحدد لون نمط التعبئة. وفيما يلي الأنماط المتوفرة في لغة سي. وكل نمط يقابله رقم مرتبةً بدءاً من الصفر إلى 12على التوالي كما يلي، أما بالنسبة للألوان فتم توضيحها سابقاً.

EMPTY_FILL(0), SOLID_FILL(1), LINE_FILL(2), LTSLASH_FILL(3), SLASH_FILL(4), BKSLASH_FILL(5), LTBKSLASH_FILL(6), HATCH_FILL(7), XHATCH_FILL(8), INTERLEAVE_FILL(9), WIDE_DOT_FILL(10), CLOSE_DOT_FILL(11), USER_FILL(12)

مثال:

لو أردنا رسم مستطيل أخضر معبأ بنمط SOLID_FILL أحمر. أحداثيات الزاوية العلوية اليسرى (100, 100) وإحداثيات الزاوية السفلية اليمنى (300, 100).

الحل:

نكتب الجمل التالية:

```
setcolor(2);
setfillstyle(SOLID_FILL, RED);
rectangle(100, 100, 300, 200);
floodfill(100, 100, 2);
```

تدريب 15:

اكتب برنامجاً بلغة سي لرسم دائرة مركزها النقطة (100, 100) ونصف قطرها 50 بلون أبيض ومعبأة بشكل النمط XHATCH_FILL ولون أحمر.

الحل:

```
#include<graphics.h>
#include<conio.h>
 main()
{  int gd = DETECT, gm;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
  setcolor(15);   //        أو بدونها لأن اللون الأبيض هو اللون الإفتراضي
  setfillstyle(XHATCH_FILL, RED);   //  setfillstyle(8, RED);
  circle(100, 100, 50);
  floodfill(100, 100, WHITE);
   getch();
  closegraph();
  return 0;    }
```

14.2 دالة setlinestyle()

تستخدم هذه الدالة لتعيين نوع خط الرسم وسماكة خط الرسم ونمطه في حال استخدام النوع USERBIT_LINE فقط. وتكتب قبل دالة الرسم.

الصيغة العامة لهذه الدالة هي:

```
void setlinestyle( int linestyle, unsigned pattern, int thickness );
```

معاملات هذه الدالة هي نمط الخط linestyle يمكن أن يكون أحد القيم التي يوفرها مترجم تيربو سي وهي خمسة (5) أنماط، وكل نمط يقابله رقم يدل عليه SOLID_LINE(0), DOTTED_LINE(1), CENTER_LINE(2), DASHED_LINE(3) USERBIT_LINE(4) المعامل الثاني يستخدم فقط مع نمط الخط user defined والمعامل الثالث يمثل سماكة الخط ويمكن أن يكون عرضها عرض 1 pixel or 3 pixel. حيث أن الواحد خط عادي والثلاث خط سميك.

مثال:

لو أردنا رسم خط متقطع بسماكة رقمها واحد وكذلك إذا أردنا رسم خط منقط بسماكة رقمها 3.

الحل:

نكتب الجمل الآتية:

```
setlinestyle(DASHED_LINE, 1, 1);
line(left + 20, top + 20, left + 70, top + 70);
setlinestyle(DOTTED_LINE, 1, 3);
line(left + 70, top + 20, left + 20, top + 70);
```

تدريب 16:
اكتب برنامجاً بلغة سي لرسم عدة خطوط تحت بعضها كل خط بنمط من الأنماط المتوفرة طول الخط 200 pixel

الحل:

```
#include <graphics.h>
#include<conio.h>
main()
{  int gd = DETECT, gm, c , x = 100, y = 50;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
   for ( c = 0 ; c < 5 ; c++ )
  {
     setlinestyle(c, 0, 2);
      line(x, y, x+200, y);
     y = y + 25;
  }
   getch();
  closegraph();
  return 0;   }
```

15.2 دالة outtext()

تستخدم هذه الدالة لكتابة نص في المكان الحالي على شاشة الرسم (مكان وجود المشيرة) ويمكن إزاحة المشيرة من مكانها إلى مكان جديد قبل الكتابة باستخدام الدالة moveto(). ولتغيير نوع الخط نستخدم الدالة settextstyle().

الصيغة العامة لهذه الدالة هي:

```
void outtext(char *string);
```

معامل هذه الدالة هو النص المراد طباعته على شاشة الرسم.

مثال:
لو أردنا كتابة العبارة Hello World على شاشة الرسم عند الإحداثيات الحالية للمشيرة، الافتراضي هو (.، .) بلون أزرق.

الحل:
نكتب الجمل الآتية:

```
setcolor(1);
outtext("Hello World");
```

ملاحظة:
لا تستخدم الأمر printf أو الرمز '\n' على شاشة الرسم.

تدريب 17:
اكتب برنامجاً بلغة سي لطباعة نص في مكان تواجد المشيرة على شاشة الرسم ثم أعد كتابة النص عند إحداثيات منتصف الشاشة.

الحل:

```
#include<graphics.h>
#include<conio.h>
main()
{  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
   outtext("To display text at a particular position on the screen use outtextxy");
  moveto(320, 240);
  outtext("To display text at a particular position on the screen use outtextxy");
```

```
  getch();
  closegraph();
  return 0;    }
```

16.2 دالة outtextxy()

تستخدم هذه الدالة لكتابة نص على شاشة الرسم في مكان يتم تحديده بإحداثيات X,Y
الصيغة العامة لهذه الدالة هي:

```
void outtextxy(int x, int y, char *string);
```

معاملات هذه الدالة المتغيرات X, Y وهي إحداثيات نقطة الكتابة على شاشة الرسم والمتغير الثالث النص المراد كتابته على شاشة الرسم.

ملاحظة:
هذه الدالة تقوم بعمل الدالتين معاً، الدالة moveto() والدالة outtext().

مثال:
لو أردنا كتابة النص "Hello World" على شاشة الرسم في الموقع الذي إحداثياته (100, 100).
الحل:
نكتب الجملة الآتي:

```
Outtextxy(100, 100, "Hello World");
```

أو
الجمل التالية

```
int x=200, y=300;
Outtextxy(x, y, "Hello World);
```

تدريب 18:
اكتب برنامجاً بلغة سي لكتابة النص "outtextxy function" على شاشة الرسم بلون أحمر عند النقطة التي إحداثياتها (100, 100) باستخدام الدالة outtextxy().

الحل:

```
#include<graphics.h>
#include<conio.h>
main()
{
  int gd = DETECT, gm;

  initgraph(&gd,&gm,"C:\\TC\\BGI");
  setcolor(4);
  outtextxy(100, 100, "Outtextxy function");
  getch();
  closegraph();
  return 0;
}
```

17.2 الدالة settextstyle()

تستخدم هذه الدالة لتغيير شكل النص (مظهره)، بواسطتها يمكن تغيير نوع الخط واتجاه كتابة النص وحجم خط النص.
الصيغة العامة لهذه الدالة هي:

```
void settextstyle( int font, int direction, int charsize);
```

يوجد للدالة ثلاث معاملات هي: المتغير font لتحديد نمط (نوع) الخط ويعبر عنه برقم من 0-10 أو بكلمات أما المتغير direction لتحديد إتجاه النص (أفقي/عمودي) ويعبر عنه برقم 0،1 أو بكلمات أما المتغير charsize لتحديد حجم الخط font size ويعبر عنه برقم من 0-10 أو بكلمات.

المتغير direction ويأخذ قيمتين هي:(Left to right) HORIZ_DIR و (Bottom to top) VERT_DIR
المعامل font ويأخذ أحد القيم الآتية:

DEFAULT_FONT(0), TRIPLEX_FONT(1), SMALL_FONT(2), SANS_SERIF_FONT(3), GOTHIC_FONT(4), SCRIPT_FONT(5), SIMPLEX_FONT(6), TRIPLEX_SCR_FONT(7), COMPLEX_FONT(8), EUROPEAN_FONT(9), BOLD_FONT(10)

القيمة الافتراضية (التلقائية) للمعاملات هي DEFAULT_FONT, HORIZ_DIR and font size of 1
مثال:
لو أردنا كتابة العبارة : "Welcome Back" بشكل أفقي وبحجم رقمه 7 وخط نوعه SCRIPT_FONT على شاشة الرسم عند النقطة التي إحداثياتها (100, 100).
الحل:
نكتب الجمل التالية:

```
settextstyle(SCRIPT_FONT, HORIZ_DIR, 7);
    outtextxy(100,100,"Welcome Back");
```

تدريب 19:
اكتب برنامجاً لكتابة العبارة "Text with different fonts" بشكل أفقي بأنواع خطوط مختلفة وأحجام مختلفة. عند النقاط التي إحداثياتها (X, Y) .

الحل:
يمكن استخدام الدالة outtextxy() أو الدالة moveto() مع الدالة outtext()

```
#include <graphics.h>
#include <conio.h>
main()
{     int gd = DETECT, gm, x = 25, y = 25, font;
   initgraph(&gd,&gm,"C:\\TC\\BGI");
   for (font = 0; font <= 10; font++)
  {
    settextstyle(font, HORIZ_DIR, font);
    outtextxy(x, y, "Text with different fonts");
    y = y + 25;
  }
   getch();
  closegraph();
  return 0;  }
```

18.2 دالة moveto()

تستخدم هذه الدالة لنقل المشيرة من موقعها الحالي على شاشة الرسم إلى مكان آخر على شاشة الرسم يحدد بقيم إحداثيات (x,y) جديدة.
الصيغة العامة لهذه الدالة هي:

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

معاملات هذه الدالة تمثل إحداثيات النقطة الجديدة على محور سين ومحور صاد.
مثال:
لو أردنا كتابة العبارة Computer Graphics على الشاشة عند الإحداثيات (100, 100)
الحل:
نكتب الجمل الآتية:

```
Moveto(100, 100);
Outtext("Computer Graphics");
```

تدريب 20:
اكتب برنامجا بلغة سي لكتابة اسم الجامعة "Alquds Open University" بلون أبيض داخل مستطيل لونه أحمر ومعبأ بلون أزرق وأحداثيات الزاوية العلوية اليسرى (100, 100) وإحداثيات الزاوية السفلية اليمنى (300, 200).

الحل:

```
#include <graphics.h>
#include <conio.h>
main()
{  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
   setcolor(RED);
  setfillcolor(SOLID_FILL, BLUE);
  rectangle(100, 100, 300, 200)
  floodfill(110, 110, 4);
  moveto(110, 140);
  setcolor(15);
  outtext("Alquds Open University");
   getch();
  closegraph();   return 0;   }
```

19.2 دالة lineto()

تستخدم هذه الدالة لرسم خط من النقطة الحالية (مكان وجود المشيرة) إلى نقطة أخرى جديدة على شاشة الرسم يتم تحديد مكانها من خلال معاملات الدالة (X, Y). وتستخدم مع دالة moveto() لتحديد النقطة الحالية.
الصيغة العامة لهذه الدالة هي:

```
void lineto (int X, int Y)
```

معاملات هذه الدالة هي المتغيرات (X,Y) وهي إحداثيات النقطة الجديدة نقطة نهاية الخط المراد رسمه.
لمعرفة إحداثيات النقطة الحالية نستخدم الدوال getx(), gety().
مثال:
لو أردنا رسم خط مستقيم من النقطة الحالية (100, 100) إلى نقطة منتصف الشاشة (320, 240) باستخدام الدوال lineto(), moveto().
الحل:
اكتب الجمل الآتية:

```
moveto(100, 100);
lineto(320, 240);
```

تدريب 21:
اكتب برنامجاً بلغة سي لرسم مستطيل بلون أحمر باستخدام الدوال lineto(), moveto() أحداثياته من الزاوية العلوية اليسرى (100, 100) وإحداثياته من الزاوية السفلية اليمنى (280, 200). ثم اكتب العبارة "done by linto()" بلون أصفر. في منتصف المستطيل.

الحل:

```
#include<graphics.h>
#include<conio.h>
 main()
{  int gd = DETECT, gm;
     initgraph(&gd, &gm, "C:\\TC\\BGI");
     setcolor(14);
     moveto(120, 150);
     outtext("done by lineto()");
     setcolor(4);
```

```
    moveto(100, 100);
    lineto(280, 100);
    lineto(280, 200);
    lineto(100, 200);
    lineto(100, 100);
  getch();
 closegraph();   return 0;}
```

20.2 دالة linerel()

تستخدم هذه الدالة لرسم خط على شاشة الرسم من النقطة الحالية إلى نقطة أخرى تبعد عن النقطة الحالية مسافة محددة على محور السينات ومسافة محددة على محور الصادات ويتم نقل المشيرة إلى النقطة الجديدة لتصبح النقطة الحالية. يمكن استخدام الدوال getx, gety لمعرفة النقطة الحالية.

الصيغة العامة لهذه الدالة هي:

```
void linerel(int x, int y);
```

معاملات هذه الدالة هي المسافة على محور السينات ومحور الصادات من النقطة الحالية. وتستخدم مع moverel.

مثال:

لو أردنا رسم خطاً من النقطة الحالية (250, 250) إلى النقطة التي إحداثياتها تنقص بمقدار (100) من النقطة الحالية على المحور Y وتزيد بمقدار (100) على المحور X. النقطة الحالية تحدد باستخدام الدالة moveto() أو الدالة lineto . ويعرف هذا المقدار بمقدار الإزاحة Offsetx, Offsety ويتم حساب نقطة نهاية الخط كما يلي:

endx = begx + offsetx;
endy = endx + offsety;

الحل:

نكتب الأوامر التالية:

```
moveto(250, 250);
linerel(100, -100);
```

مثال:

ما هي احداثيات الخط الذي يتم رسمه بواسطة الجمل التالية:

```
Moveto(20, 30);
moverel(100, 30);
linerel(320, 240);
```

الحل:

إحداثيات الخط هي (120,60), (320, 270)

تدريب 22:

اكتب برنامجاً بلغة سي لرسم مربع بلون أصفر طول ضلعه 50 مبتدأً من النقطة (100, 100). باستخدام الدالة linerel()

الحل:

```
#include <graphics.h>
#include <conio.h>
 main()
{  int gd = DETECT, gm;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
    setcolor(YELLOW);
    moveto(100, 100);
    linerel(50, 0);
    linerel(0, 50);
    linerel(-50, 0);
    linerel(0, -50);
     getch();
  closegraph();
  return 0;}
```

21.2 دالة moverel()

تستخدم هذه الدالة لتحريك أو نقل المشيرة من مكان إلى آخر على شاشة الرسم بمسافة محددة بإتجاه محور السينات أو محور الصادات أو كلاهما معاً.
الصيغة العامة لهذه الدالة هي:

```
void moverel(int x, int y);
```

معاملات هذه الدالة هي المسافة على محور السينات والمسافة على محور الصادات.

ملاحظة:
يمكن معرفة إحداثيات النقطة الحالية باستخدام الدوال getx(), gety() كما يلي:

```
x = getx();
y = gety();
```

مثال:
لو أردنا نقل المشيرة من النقطة الحالية (100, 100) إلى النقطة التي إحداثياتها (150, 50) باستخدام الدالة moverel() .
الحل:
نكتب الجمل الآتية:

```
moveto(100, 100);
moverel(50, -50);
```

تدريب 23:
اكتب برنامجاً بلغة سي لكتابة العبارة "Moveto Example" عند النقطة التي إحداثياتها (100, 100). ثم أنقل المشيرة إلى موقع جديد يتمثل بالنقطة التي إحداثياتها (100, 150) باستخدام الدالة moverel() ثم اكتب العبارة "Moverel Example"

الحل:

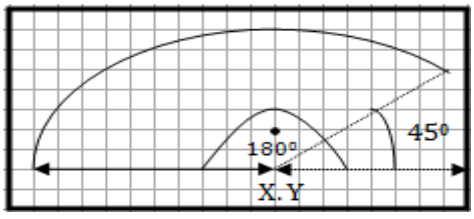
```
#include <graphics.h>
#include <conio.h>
 main()
{   int gd = DETECT, gm, x, y;
    initgraph(&gd, &gm, "C:\\TC\\BGI");
   moveto(100, 100);
  outtext("Moveto Example");
  moverel(0, 50);
  outtext("Moverel Example");
   getch();
  closegraph();
  return 0;}
```

22.2 دالة arc()

تستخدم هذه الدالة لرسم أقواس دائرية، إحداثيات نقطة المركز (x, y) على شاشة الرسم.
الصيغة العامة لهذه الدالة هي:

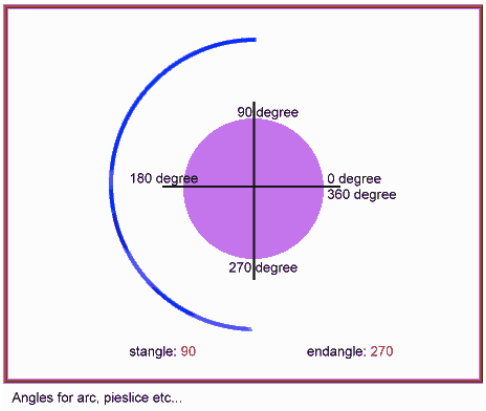
```
void arc(int x, int y, int stangle, int endangle, int radius);
```

معاملات هذه الدالة هي نقطة المركز وإحداثياتها (x, y) وزاوية بداية الرسم stangle وزاوية نهاية الرسم endangle ونصف قطر القوس radius.



ويمكن استخدام هذه الدالة لرسم دائرة بحيث تكون زاوية البداية 0 وزاوية النهاية 360.
الشكل التالي يوضح نقطة مركز القوس وهي (X, Y) وزاوية البداية وهي 45^0 وزاوية النهاية وهي 180^0

مثال:
لو أردنا رسم الأقوس التالية:
قوس مركزه (100, 100) وزاوية البداية صفر وزاوية النهاية 90 ونصف قطره 35
قوس مركزه (100, 100) وزاوية البداية 90 وزاوية النهاية 180 ونصف قطره 35
قوس مركزه (100, 100) وزاوية البداية 180 وزاوية النهاية 270 ونصف قطره 35
قوس مركزه (100, 100) وزاوية البداية 270 وزاوية النهاية 360 ونصف قطره 35



الحل:
نكتب الجمل التالية:

```
arc(100, 100, 0, 90, 35);
arc(100, 100, 90, 180, 35);
arc(100, 100, 180, 270, 35);
arc(100, 100, 270, 360, 35);
```

تدريب 24:
اكتب برنامجاً بلغة سي لرسم قوس مركزه النقطة (320, 240) وزاوية البداية 0 وزاوية النهاية 45 ونصف قطره 100.

الحل:

```
#include <graphics.h>
#include <conio.h>
 main()
{   int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");
    arc(320, 240, 0, 45, 100);
    getch();
   closegraph();   return 0;    }
```

23.2 دالة bar()

تستخدم لرسم مستطيل ذو بعدين (شريط) ممتلئ. بدون حدود للشكل. ولوضع حدود نستخدم الدالة rectangle() مع الدالة bar(). هذه الدالة تشبه دالة المستطيل ولكن هذه الدالة تقوم بتعبئة المستطيل بالنمط واللون الحالي وهو الإفتراضي. ولتغيير نمط التعبئة ولون التعبئة نستخدم الدالة setfillstyle.

الصيغة العامة لهذه الدالة هي:

```
void bar(int left, int top, int right, int bottom);
```

معاملاتها نفس معاملات المستطيل نحدد إحداثيات الزاوية العلوية اليسرى وإحداثيات الزاوية السفلية اليمنى.

مثال:
لو أردنا رسم شريط أزرق إحداثياته يتم حسابها بناءً على إحداثيات الشاشة وإعطاء مسافة للهامش margin ووضع إطار له باستخدام الدالة rectangle().

الحل:
نكتب الجمل التالية:

```
margin = 100; // 100 pixel margin
left = margin;
top = margin;
bottom = getmaxy() – margin;
```

```
right = getmaxx() – margin;
// draw a bar with solid fill on blue color  and   use rectangle to have a white color border
setfillstyle(SOLID_FILL, BLUE);
bar(left, top, right, bottom);
rectangle(left, top, right, bottom);
```

تدريب 25:
اكتب برنامجاً بلغة سي يرسم شريط (مستطيل ممتلئ بدون حدود) إحداثيات الزاوية العلوية اليسرى (100, 150) وإحداثيات الزاوية السفلية اليمنى (200, 200).

الحل:

```
#include <graphics.h>
#include <conio.h>
main()
{  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  bar(100, 150, 200, 200);        شريط طوله 100 وعرضه 50 معبأ بلون أبيض وهو اللون الإفتراضي//
  getch();
  closegraph();
  return 0;   }
```

24.2 دالة bar3d()

تستخدم لرسم شريط (مستطيل) ثلاثي الأبعاد ولهذا فإن الدالة لها إحداثيات للبعد الثالث وتعبأ الوجه الأمامي فقط. ولتعبئة الوجه العلوي والوجه الجانبي نستخدم الدالة floodfill(). الدالة floodfill() تعبئ باللون الذي يتم تحديدة بواسطة الدالة setfillstyle()

الصيغة العامة لهذه الدالة هي:

```
void bar3d(int left, int top, int right, int bottom, int depth, int topflag);
```

المعاملات الجديدة هي depth وتمثل العمق للشكل أما المعامل topflag قيمته تحدد إذا للشكل وجه علوي أم لا, إذا كانت قيمتها أكبر من صفر يكون للشكل وجه علوي وإذا كانت صفر يرسم الشكل بدون الوجه العلوي.

مثال:

أرسم شكلاً ثلاثي الأبعاد (شريط) مستطيل له عمق مقداره 35 بكسل ليصبح شكله يشبه الصندوق. ولون الواجه الأمامية أزرق والواجه العلوية لونها أصفر والجانب لونه أخضر . إحداثيات الزاوية العلوية اليسرى (100, 100) وإحداثيات الزاوية السفلية اليمنى (200, 150)

الحل:

```
            //draw a 3d bar
    setfillstyle(SOLID_FILL, BLUE);
    bar3d(100, 100, 200, 150, 35, 1);     // floodfill with YELLOW color
    setfillstyle(SOLID_FILL, YELLOW);
    floodfill(150, 90, WHITE);     // floodfill with GREEN color
    setfillstyle(SOLID_FILL, GREEN);
    floodfill( 210, 125, WHITE);
```

تدريب 26:
اكتب برنامجاً بلغة سي يرسم مستطيلاً ثلاثي الأبعاد إحداثيات الزاوية العلوية اليسرى (100, 100) وإحداثيات الزاوية السفلية اليمنى (200, 200) وعمقه 20.

الحل:

```
#include<graphics.h>
#include<conio.h>
main()
{  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
```

```
 bar3d(100, 100, 200, 200, 20, 1);
getch();
closegraph();
return 0;    }
```

تدريب 27:
اكتب برنامجاً بلغة سي يرسم 12 مستطيلاً ثلاثي الأبعاد ويعبأ كل مستطيل بنمط ولون مختلفين.

الحل:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
void main()
{
        int gd=DETECT, gm,bkcolor;

        initgraph(&gd,&gm,"c:\\tc\\bgi");
        setfillstyle(EMPTY_FILL,YELLOW);
        bar3d(2,150,100,200,25,1);
        setfillstyle(SOLID_FILL,RED);
        bar3d(150,150,250,200,25,1);
        setfillstyle(LINE_FILL,BLUE);
        bar3d(300,150,400,200,25,1);
        setfillstyle(LTSLASH_FILL,GREEN);
        bar3d(450,150,550,200,25,1);
        setfillstyle(SLASH_FILL,CYAN);
        bar3d(2,250,100,300,25,1);
        setfillstyle(BKSLASH_FILL,BROWN);
        bar3d(150,250,250,300,25,1);
        setfillstyle(LTBKSLASH_FILL,MAGENTA);
        bar3d(300,250,400,300,25,1);
        setfillstyle(HATCH_FILL,LIGHTRED);
        bar3d(450,250,550,300,25,1);
        setfillstyle(XHATCH_FILL,DARKGRAY);
        bar3d(2,350,100,400,25,1);
        setfillstyle(INTERLEAVE_FILL,YELLOW);
        bar3d(150,350,250,400,25,1);
        setfillstyle(WIDE_DOT_FILL,LIGHTMAGENTA);
        bar3d(300,350,400,400,25,1);
        setfillstyle(CLOSE_DOT_FILL,LIGHTGRAY);
        bar3d(450,350,550,400,25,1);
        getch();
        closegraph();     }
```

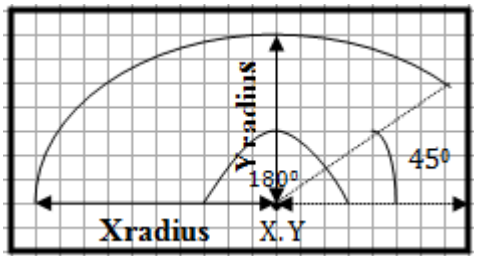
25.2 دالة ellipse()

تستخدم هذه الدالة لرسم شكل بيضاوي (قطع ناقص) مركزه (x, y) على شاشة الرسم.
الصيغة العامة لهذه الدالة هي:

```
void ellipse(int x, int y, int stangle, int endangle, int xradius, int yradius);
```

معاملات هذه الدالة هي: إحداثيات نقطة مركز الشكل ممثلة بقيم (x, y) وزاوية بداية الرسم وزاوية نهاية الرسم ونصف القطر على محور السينات ونصف القطر على محور الصادات.
ملاحظة:

لرسم الشكل البيضاوي كاملاً (مغلق) زاوية البداية يجب أن تكون صفر وزاوية النهاية يجب أن تكون 360. وإلا يكون الشكل قوساً.
الشكل التالي يوضح نقطة مركز القوس وهي (X, Y) وزاوية البداية وهي 45^0 وزاوية النهاية وهي 180^0 ونصف القطر على محور السينات Xradius ونصف القطر على محور الصادات Yradius



مثال:
ما هو الشكل الذي نحصل عليه من الأمر الآتي: ellipse(320, 240, 0, 240, 50, 100);
الحل:

مثال:
لو أردنا رسم شكل بيضاوي بلون أزرق مركزه (200, 200) ونصف قطره على محور السينات 100 ونصف قطره على محور الصادات 60
الحل:
نكتب الجمل الآتية:

```
setcolor(BLUE);
ellipse(200, 200, 0, 360, 100, 60);
```

مثال:
لو أردنا رسم قوس arc أبيض إحداثيات مركزه (150, 150) وزاوية بداية الرسم 270 وزاية نهاية الرسم 360 ونصف قطره على محور السينات 100 ونصف قطره على محور الصادات 70 باستخدام الدالة ellipse. نهايات القوس مرتبطة بمركزه بخطوط مستقيمة. ثم اكتب العبارة التالية أعلى الشكل: "Ellipse 270-360 deg".
الحل:

```
setcolor(YELLOW);
 moveto(150, 150);
 linerel(100, 0);
 moveto(150, 150);
 linerel(0,70);
 ellipse(150, 150, 270, 0, 100, 70);
 outtextxy(140, 130, "Ellipse 270-360 deg");
```

تدريب 28:
اكتب برنامجاً بلغة سي لرسم شكل بيضاوي إحداثيات مركزه (100, 100) ونصف قطره على محور السينات 50 ونصف قطره على محور الصادات 25 .

الحل:

```
#include<graphics.h>
#include<conio.h>
main()
{  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
   ellipse(100, 100, 0, 360, 50, 25);
  getch();
  closegraph();
  return 0;   }
```

26.2 دالة fillellipse()

تستخدم هذه الدالة لرسم شكل بيضاوي وتعبئته باللون الإفتراضي الأبيض أو تعبئتها بأحد أنماط التعبئة ولون آخر والتي يتم تحديدها بالدالة setfillstyle().

الصيغة العامة لهذه الدالة هي:

```
void fillellipse(int x, int y, int xradius, int yradius);
```

معاملات هذه الدالة هي إحداثيات نقطة المركز للشكل البيضاوي (x, y) ونصف القطر على محور السينات ونصف القطر على محور الصادات.

مثال:

لو أردنا رسم شكل بيضاوي وتعبئته بلون أزرق من النمط SOLID_FILL إحداثيات المركز هي (120, 120) ونصف قطره على محور السينات 90 ونصف قطره على محور الصادات 50 وبداخلها العبارة "fillellipse 0-360 deg"

الحل:

```
setfillstyle(SOLID_FILL, BLUE);
fillellipse(120, 120, 90, 50);
outtextxy(35, 120, "fillellipse 0-360 deg");
```

تدريب 29:

اكتب برنامجاً بلغة سي لرسم شكل بيضاوي مع بأباللون والنمط الافتراضيين.

الحل:

```
#include <graphics.h>
#include <conio.h>
 int main()
{  int gd = DETECT, gm;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
   fillellipse(100, 100, 50, 25);
   getch();
  closegraph();
  return 0;    }
```

27.2 دالة drawpoly()

تستخدم هذه الدالة لرسم شكل متعدد الأضلاع (مضلع) ثلاثة أضلاع او أكثر مثل المثلث، المربع، المستطيل، الشكل الخماسي ..... الخ. باللون والنمط الإفتراضيين.

الصيغة العامة لهذا الشكل هي:

```
void drawpoly( int num, int *polypoints );
```

معاملات هذه الدالة هي عدد رؤوس الشكل + 1 المعامل الثاني سلسلة أرقام تمثل إحداثيات الرؤوس حيث أن كل زوج من الأرقام يمثل إحداثيات نقطة على محور سين ومحور صاد (رأس واحد من رؤوس المضلع) إذاً عدد الأرقام يساوي ((عدد رؤوس الشكل +1)* 2). لأن من أجل رسم شكل مغلق بعدد n من الرؤوس يجب تمرير n+1 من الإحداثيات إلى الدالة drawpoly حيث أن آخر إحداثيات تساوي أول إحداثيات.

drawpoly(number_of_vertices+1 , sequence_of_polygonpoints)

مثال:

لو أردنا رسم مثلث باستخدام الدالة drawpoly() إحداثيات رؤوسه (420,250), (420, 350), (250,300)

الحل:

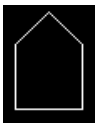
نكتب الجمل التالية:

```
points[]={420,250,420,350,250,300,420,250};
drawpoly(4, points);
```

ويمكن كتابتها أيضاً بالشكل التالي:

```
Points[0]=420;points[1]=250;points[2]=420;points[3]=350;points[4]=350;points[5]=300;
points[6]=420;points[7]=250;
drawpoly(4,points);
```

مثال:

لو أردنا أن رسم شكل المضلع  **الآتي:**

مضلع بخمس أضلاع Pentagon باستخدام الدالة drawpoly نضع إحداثيات رؤوس المضلع في المصفوفة array[] كما يلي:

الحل:

نكتب الجمل التالية:

```
int array[]={320,240,340,220,360,240,360,280,320,280,320,240};
drawpoly(6,array);
```

تدريب 30:

اكتب برنامجاً بلغة سي لرسم مثلث باستخدام الدالة drawpoly() إحداثيات رؤوس المثلث ه ي (320,150,420, 300, 250, 300)

الحل:

```
#include <graphics.h>
#include <conio.h>
 main()
{
  int gd=DETECT,gm,points[]={320,150,420,300,250,300,320,150};
   initgraph(&gd, &gm, "C:\\TC\\BGI");
   drawpoly(4, points);
   getch();
  closegraph();
  return 0;   }
```

28.2 دالة fillpoly()

تستخدم هذه الدالة لرسم شكل متعدد الأضلاع (مضلع) وتعبئته باللون والنمط الحاليين (الافتراضيين) ويمكن تغييرها باستخدام الدالة setfillstyle.

الصيغة العامة لهذه الدالة هي:

```
void fillpoly( int num, int *polypoints );
```

معاملات هذه الدالة نفس معاملات الدالة drawpoly `fillpoly( number_of_vertices+1 , sequence_of_polygonpoints )`

مثال:

لو أردنا أن نرسم مثلث معبأ باللون والنمط الافتراضيين وإحداثيات رؤوس المثلث هي (420,250), (420, 350), (250,300)

الحل:

نكتب الجمل الآتية:

نعرف مصفوفة تحتوي على إحداثيات رؤوس المثلث: `points[]={320,150,440,340,230,340,320,150};`

نكتب أمر الرسم والتعبئة: `fillpoly(4, points);`

مثال:

لو أردنا أن نرسم الشكل  **الآتي:**

الحل:

نكتب الجمل التالية:

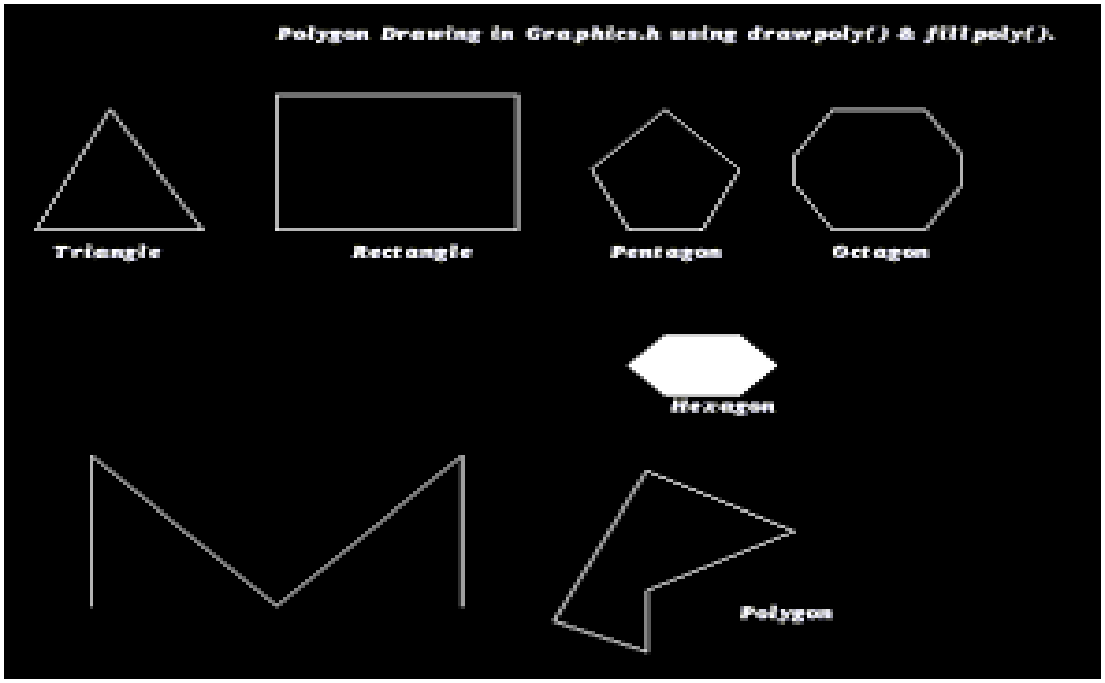
```
array[]={320,240,340,220,360,240,360,280,320,280,320,240};
fillpoly(6,array);
```

تدريب 31:
اكتب برنامجاً بلغة سي لرسم شكل أبيض معبأ بلون أحمر وبالنمط SOLID_FILL إحداثيات رؤوسه هي (540,220),(490,270),(510,320),(570,320),(590,270),(540,220) ثم اكتب كلمة "POLY" عند الاحداثيات (515,270)

الحل:

```
#include <graphics.h>
#include <conio.h>
 main()
{
  int gd=DETECT,gm,points[]={540,220,590,270,570,320,510,320,490,270,540,220};
   initgraph(&gd, &gm, "C:\\TC\\BGI");
    setcolor(WHITE);
    setfillstyle(SOLID_FILL, RED);
    fillpoly(6, points);
    outtextxy(515,270,"POLY");
   getch();
  closegraph();
  return 0;
}
```

تدريب 32:
اكتب برنامجاً بلغة سي لرسم الأشكال الموضحة في الشكل التالي:



الحل:

```
#include<graphics.h>
int main()
{
   int gd=DETECT, gm;
   int triangle[8]={20,150, 60,70, 110,150, 20,150};
   int rect[10]={150,60, 280,60, 280,150, 150,150, 150,60};
   int pentagon[12]={340,150, 320,110, 360,70, 400,110, 380,150, 340,150};
   int hexagon[14] ={ 360,260, 340,240, 360,220, 400,220, 420,240, 400,260, 360,260};
```

```
    int octagon[18]={450,150, 430,120, 430,100, 450,70, 500,70, 520,100, 520,120, 500,150, 450,150};
    int poly1[10]={50,400, 50,300, 150,400, 250,300, 250,400 };
    int poly2[12]={350,430, 350,390, 430,350, 350,310, 300,410, 350,430 };

    initgraph(&gd, &gm,NULL);

    outtextxy(150,15, "Polygon Drawing in Graphics.h using drawpoly() & fillpoly().");
    drawpoly(4,triangle);
    outtextxy(30,160, "Triangle");
    drawpoly(5,rect);
    outtextxy(190, 160, "Rectangle");
    drawpoly(6,pentagon);
    outtextxy(330, 160, "Pentagon");
    drawpoly(9,octagon);
    outtextxy(450, 160, "Octagon");
    fillpoly(7,hexagon);
    outtextxy(362, 262, "Hexagon");
    drawpoly(5,poly1);
    drawpoly(6,poly2);
    outtextxy(400, 400, "Polygon");
    getch();
    closegraph();
    return 0;
}
```

29.2 دالة pieslice()

تستخدم هذه الدالة لرسم دائرة أو جزءاً من دائرة (قوساً) معبأ بلون ونمط تعبئة يتم تحديده باستخدام الدالة **setfillstyle** وإذا لم يحدد يتم استخدام اللون والنمط الافتراضي وهو اللون الأبيض.

الصيغة العامة لهذه الدالة هي:

```
Pieslice(midx,midy,stangle,endangle,radius);
```

معاملات هذه الدالة إحداثيات نقطة المركز (x, y) وزاوية بداية الرسم **stangle** وزاوية نهاية الرسم **endangle** ونصف القطر **radius**

ملاحظة:

إذا كانت زاوية بداية الرسم صفراً (0) وزاوية نهاية الرسم 360 درجة نحصل على دائرة.

مثال:

لو أردنا أن نرسم شكلاً يمثل جزءاً من دائرة معبأ بلون أخضر إحداثيات مركزه (320, 240) وزاوية بداية الرسم **stangle** تساوي صفر(0) وزاوية نهاية الرسم **endangle** تساوي 130 ونصف قطره **radius** يساوي 50.

الحل:

نكتب الجمل الآتية:

```
Setfillstyle(SOLID_FILL, GREEN);
Pieslice (320, 240, 0, 135, 50);
```

مثال:

لو أردنا أن نرسم الشكل الآتي:

الحل:

نكتب الجملة التالية:

```
pieslice(320,240,0,75,100);
```

تدريب 33:

اكتب برنامجاً بلغة سي لرسم شكلاً يمثل جزءاً من دائرة معبأ بلون أبيض نصف قطره 100 وزاوية بداية الرسم 45 وزاوية نهاية الرسم 225 وإحداثيات نقطة المركز (200, 200).

الحل:

```
#include <graphics.h>
#include <conio.h>
 main()
{  int gd = DETECT, gm;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
          pieslice(200, 200, 45, 225, 100);
   getch();
  closegraph();
  return 0;    }
```

30.2 دالة sector()

تستخدم هذه الدالة لرسم شكلاً بيضاوياً أو جزءاً من شكلاً بيضاوياً معبأ بلون أو نمط تعبئة يتم تحديده باستخدام الدالة setfillstyle() وإذا لم يحدد اللون يتم استخدام اللون الأبيض الإفتراضي.
الصيغة العامة لهذه الدالة هي:

```
void sector( int x, int y, int stangle, int endangle, int xradius, int yradius);
```

معاملات هذه الدالة نقطة مركز رسم الشكل (x, y) وزاوية بداية الرسم وزاوية نهاية الرسم ونصف قطر محور السينات ونصف قطر محور الصادات. إذا كانت زاوية الهداية (0) وزاوية النهاية (360) نحصل على شكل بيضاوي كاملاً مع أ.

مثال:
لو أردنا رسم جزاً من شكلاً بيضاوياً معبأ بلون أخضر وبنمط تعبئة CLOSE_DOT_FILL إحداثيات نقطة مركزه (100, 100) وزاوية بداية الرسم (0)وزاوية نهاية الرسم (270) ونصف قطر محور السينات (40) ونصف قطر محور الصادات (25). ثم اكتب العبارة sector 0-270 deg عند إحداثيات النقطة (70,80)

الحل:
نكتب الجمل الآتية:

```
 setfillstyle(CLOSE_DOT_FILL, GREEN);
sector(100, 100, 0, 270, 40, 25);
outtextxy(70, 80, "sector 0-270 deg");
```

مثال:
لو أردنا أن نرسم الشكل  **الآتي:**
الحل:
نكتب الجملة التالية:

```
sector(320,240,0,180,100,50);
```

تدريب 34:
اكتب برنامجاً بلغة سي لرسم شكلاً يمثل جزءاً من شكلاً بيضاوياً إحداثيات نقطة مركزه (100, 100) وزاوية بداية الرسم (0)وزاوية نهاية الرسم (135) ونصف قطر محور السينات (25) ونصف قطر محور الصادات (35) معبأ باللون الإفتراضي الأبيض.

الحل:

```
#include <graphics.h>
#include <conio.h>
 main()
{  int gd = DETECT, gm;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
   sector(100, 100, 0, 135, 25, 35);
   getch();
  closegraph();
  return 0;   }
```

31.2 دالة getx()

تستخدم للحصول على أو إعادة مكان المشيرة على شاشة الرسم (الإحداثيات السينية للموقع الحالي للمشيرة) ويمكن تغييره باستخدام الدوال moveto(), moverel()
الصيغة العامة لهذه الدالة هي:

```
int getx();
```

مثال:
لو أردنا معرفة الإحداثيات السينية لموقع المشيرة على شاشة الرسم
الحل:
نكتب الجملة التالية int x = getx()

32.2 دالة gety()

تستخدم للحصول على أو معرفة مكان المشيرة على شاشة الرسم (الإحداثيات الصادية للموقع الحالي للمشيرة) ويمكن تغييره باستخدام الدوال moveto(), moverel()

الصيغة العامة لهذه الدالة هي:

```
int gety();
```

مثال:

لو أردنا معرفة الإحداثيات الصادية لموقع المشيرة على شاشة الرسم.

الحل:
نكتب الجملة int y = gety()

تدريب 35:
اكتب برنامجاً بلغة سي لطباعة النص الآتي على شاشة الرسم "This is a test for getx and gety" عند النقطة التي إحداثياتها السينية 50 وإحداثياتها الصادية 70.

الحل:

```
#include <graphics.h>
#include <conio.h>
 main()
{  int gd = DETECT, gm;
    char array[100];
     initgraph(&gd, &gm, "C:\\TC\\BGI");
     moveto(50, 70);
  int x = getx();
  int y = gety();
  outtextxy(x, y, "This is a test for getx and gety");
  sprintf(array, "Current position of X = %d",x ,"Current position of Y=%d",y);
  outtextxy(50,90, array)
   getch();
  closegraph();
  return 0;   }
```

33.2 دالة getmaxx()

تستخدم هذه الدالة للحصول على أو اعادة القيمة القصوى للإحداثي السيني على شاشة الرسم. مشغل الرسم عادة يكون VGA و نمط شاشة الرسم يكون VGAHI لمشغل 639

الصيغة العامة لهذه الدالة هي:

```
int getmaxx();
```

34.2 دالة getmaxy()

تستخدم هذه الدالة للحصول على أو استرجاع القيمة القصوى لمحور صاد لشاشة الرسم. مشغل الرسم عادةً يكون 479 لمشغل VGA ونمط شاشة الرسم يكون VGAHI.

الصيغة العامة لهذه الدالة هي:

```
int getmaxy();
```

ملاحظة:

هذه الدوال مفيدة لمعرفة حدود منطقة الرسم القصوى. وتساعد في تحديد نقطة المنتصف لشاشة الرسم.

مثال:

لو أردنا رسم إطار حول منطقة الرسم لشاشة الرسم نستعين بالدوال getmaxy() ,getmaxx().

الحل:

نكتب الجملة الآتية:

```
// draw a white color border with rectangle
rectangle(0,0,getmaxx(),getmaxy());
```

تدريب 36:

اكتب برنامجاً بلغة سي لرسم نقطة صفراء في منتصف الشاشة. ومربع أزرق في منتصف الشاشة طول ضلعه 200 استخدم الدوال getmaxy() , getmaxx() في الحل ثم ارسم دائرة حمراء داخل المستطيل نصف قطرها 100 مركزها منتصف الشاشة وخط مستقيم أبيض يمثل نصف قطر الدائرة على المحور السيني:

الحل:

```
#include<graphics.h>
#include<conio.h>
 main()
{  int gd = DETECT, gm, maxx, maxy;
    initgraph(&gd,&gm,"C:\\TC\\BGI");
  maxx = getmaxx();
  maxy = getmaxy();
  int cx = (0 + maxx())/2;  //center of screen x-axis
  int cy = (0 + maxy())/2;  //center of screen y-axis
  putpixel(cx, cy, 14);
  setcolor(1);
  rectangle(cx-100, cy-100, cx+100, cy+100);
  setcolor(4);
  circle(cx, cy, 100);
  setcolor(15);
  moveto(cx, cy);
  lineto(cx+100, cy );
  getch();
  closegraph();
  return 0;   }
```

35.2 دالة textheight()

تستخدم هذه الدالة للحصول على أو إعادة إرتفاع النص بالبكسل.

الصيغة العامة لهذه الدالة هي:

```
int textheight(char *string);
```

معامل هذه الدالة النص المراد معرفة إرتفاعه.

مثال:

```
W=textwidth("graphics programming");
```

36.2 دالة textwidth()

تستخدم هذه الدالة للحصول على أو إعادة عرض النص بالبكسل.
الصيغة العامة لهذه الدالة هي:

```
int textwidth(char *string);
```

معامل هذه الدالة النص المراد معرفة عرضه:
مثال:

```
W=textheight("graphics programming");
```

ملاحظة:
استخدم الدالة settextstyle لتغيير حجم النص، ونوع الخط، واتجاه النص كما تم توضيحه سابقاً وبعدها أحصل على ارتفاع وعرض النص.

تدريب 37:
أوجد ناتج تنفيذ البرنامج التالي

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
void main()
{  int gd=DETECT, gm,w;
char arr[100];
initgraph(&gd,&gm," ");
w=textwidth("graphics programming");
sprintf(arr,"textwidth of given string=%d",w);
outtextxy(320,240,arr);
w=textheight("graphics programming");
sprintf(arr,"textheight of given string=%d",w);
outtextxy(320,260,arr);
getch();
closegraph();   }
```

الحل:
يطبع الجمل التالية:

```
textwidth of given string=160

textheight of given string=8
```

مثال:
لو أردنا كتابة النص الآتي "Palestine is land of crises" في منتصف شاشة الرسم بلون أصفر وبحجم 5 بالإتجاه الأفقي. ونوع "BOLD_FONT".

الحل:

نكتب الجمل التالية:

```
Settextstyle(BOLD_FONT, HORIZ_DIR, 5)
Width = textwidth( "Palestine is land of crises");
Height = textheight( "Palestine is land of crises");
cx = (0 + getmaxx()) / 2;
cy = (0 + getmaxy()) / 2;
setcolor(14);
outtextxy(cx-(width/2), cy-(height/2), "Palestine is land of crises");
```

تدريب 38:
اكتب برنامجاً بلغة سي لكتابة النص الآتي: "This is a test for textheight and textwidth" بلون أزرق في منتصف الشاشة داخل مستطيل معبأ بلون أحمر.

الحل:

```
#include<graphics.h>
#include<conio.h>
 main()
{   int gd = DETECT, gm, height;
    initgraph(&gd, &gm, "C:\\TC\\BGI");
    Settextstyle(TRIPLEX_FONT, HORIZ_DIR, 9)
   height = textheight("This is a test for textheight and textwidth ");
   width = textheight("This is a test for textheight and textwidth ");
   cx = (0 + getmaxx()) / 2;
   cy = (0 + getmaxy()) / 2;
   setcolor(1);
   outtextxy(cx-(width/2), cy-(height/2), "This is a test for textheight and textwidth ");
   setfillstyle(SOLID_FILL,RED);          //fill color
   rectangle(cx-70,cy-50, cx+70, cy+50);
   floodfill(cx, cy,15)          //border color
   getch();
   closegraph();
   return 0;    }
```

37.2 دالة Setviewport()

تستخدم هذه الدالة لتحديد إحداثيات منطقة عرض الرسومات (الأشكال) على الشاشة. أي لحصر عرض الرسومات على جزء من الشاشة.
الصيغة العامة لهذه الدالة هي:

```
void setviewport (int left, int top, int right, int bottom, int clip);
```

معاملات الدالة هي إحداثيات منطقة عرض الأشكال الممثلة بإحداثيات الزاوية العلوية اليسرى للمنطقة وتصبح هي نقطة الأصل (0, 0) والزاوية السفلية اليمنى للمنطقة . أما المعامل clip فهو يستخدم لتحديد إذا ما الرسومات يتم قصها (تقليمها) عند حدود منطقة العرض أم لا. إذا كانت قيمة المعامل رقم غير الصفر فإنه يتم تقليم كل الأشكال مع حدود منطقة العرض خاصة إذا كان حجم الشكل أكبر من حدود منطقة الرسم . وإذا كانت قيمة المعامل clip تساوي صفراً فإنه يتم عرض الشكل كاملاً بحيث يظهر جزءاً منه خارج منطقة العرض وإذا تم مسح منطقة العرض يبقى الجزء الخرجي ظاهراً لا يمسح.

مثال:
لو أردنا حصر منطقة الرسم بالإحداثيات (100, 100, 200, 200) لعرض الأشكال التي نرسمها داخل هذه المنطقة. أرسم دائرة نصف قطرها 55 في منتصف هذه المنطقة.

الحل:
نكتب الجمل التالية:

```
midx = getmaxx()/2;
midy = getmaxy()/2;
setviewport(midx - 50, midy - 50, midx + 50, midy + 50, 1);
circle(50, 50, 55);
```

38.2 دالة clearviewport()

تستخدم هذه الدالة لمسح الأشكال من منطقة العرض التي تم حجزها باستخدام الدالة setviewport وليس مسح كل الشاشة.
الصيغة العامة لهذه الدالة هي:

```
void clearviewport(void);
```

ملاحظة:
استخدم الدالة graphdefaults() لإلغاء المنطقة التي تم حصرها باستخدام الدالة setviewport() والعودة إلى الوضع الإفتراضي. أو باستخدام الدالة setviewport(0, 0, 640, 480); بإحداثيات الشاشة الإفتراضية.

تدريب 39:
اوجد ناتج تنفيذ البرنامج التالي:

```
# include <conio.h>
# include <stdio.h>
# include <graphics.h>
main()
{       int g_dr, g_mode;
        detectgraph(&g_dr, &g_mode);
        clrscr();
        initgraph(&g_dr, &g_mode, "c:\\tc\\bgi");
        cleardevice();
        outtextxy(0,0, "BINFO SOFTWARE");
        setviewport(100,100,150,150,1);
        getch();
        cleardevice();
        outtextxy(0,0, "BINFO SOFTWARE");
        outtextxy(10,10, "HELLO");
        getch();
        clearviewport();
        cleardevice();
        outtextxy(0,0, "BCA");
        outtextxy(10,10, "Hi");
        getch();
        closegraph();     }
```

الحل:

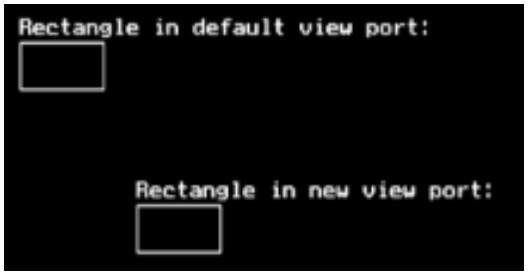
يتم كتابة العبارة"BINFO SOFTWARE" عند نقطة الأصل للشاشة وهي (0, 0) بعد ذلك يتم مسح الشاشة. ثم حصر منطقة عرض الرسومات في المنطقة المحصورة بين (100, 100) و (150, 150). وعرض العبارة السابقة عند نقطة الأصل الجديدة وهي (100, 100) وسوف يتم كتابة الأحرف BINFO فقط بسبب التقليم الذي يحدث لها لأن حجمها أكبر من المنطقة التي تم حصرها وكتابة كلمة HELLO داخل المنطقة التي تم حصرها عند النقطة التي إحداثياتها (10, 10). بعد ذلك تم مسح المنطقة المحصورة من الأشكال . ومسح الشاشة. وكتابة الأحرف "BCA" وكلمة "Hi" داخل المنطقة المحصورة.

مثال:
أوجد ناتج تنفيذ البرنامج التالي:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
```

```
void main()
 {        int gd=DETECT, gm,bkcolor;
          char a[50];
           initgraph(&gd,&gm," ");
           outtextxy(10,10,"Rectangle in default view port:");
           rectangle(10,20,60,40);
           setviewport(70,70,400,400,1);
           outtextxy(10,10,"Rectangle in new view port:");
           rectangle(10,20,60,40);
           getch();
           closegraph();    }
```

الحل:



39.2 دالة graphdefaults()

تستخدم هذه الدالة لاستعادة إعدادات الرسم إلى القيم الإفتراضية.
الصيغة العامة لهذه الدالة هي:

```
void graphdefaults();
```

إعدادات الرسم التي تسترجعها هي:

- **تعيين منفذ العرض viewport إلى الشاشة بأكملها أي إلغاء مفعول الأمر setviewport.**
- **تحريك المشيرة إلى النقطة التي إحداثياتها (0،0).**
- **ضبط لوحة الألوان الافتراضية، لون الخلفية، ولون خط الرسم.**
- **تعيين نمط التعبئة الافتراضي وشكله.**
- **تعيين نوع الخط لافتراضي ومحاذاة النص (ضبط).**

مثال:
لو أردنا وضع إعدادات الرسم إلى القيم الإفتراضية بعد أن قمنا بتغيير لون خط الرسم إلى أحمر ولون الخلفية إلى اللون الأصفر ثم رسم دائرة نصف قطرها 50 عند النقطة التي إحداثياتها (250, 250) ثم الإنتظار حتى يتم الضغط على إي زر من قبل المستخدم. بعد ذلك يتم استدعاء الدالة graphdefaults حيث تعيد لون خط الرسم ولون الخلفية إلى الأبيض والأسود.
الحل:
نكتب الجمل التالية:

```
setcolor(RED);   setbkcolor(YELLOW);
circle(250, 250, 50);   getch();
graphdefaults();
```

40.2 دالة getimage()

تستخدم هذه الدالة لنسخ جزء (مستطيل) إلى الذاكرة (حفظ جزء من صورة من منطقة محددة (مستطيل) في الذاكرة) ولها 5 معاملات إحداثيات الزاوية العلوية اليسرى وإحداثيات الزاوية السفلية اليمنى ومؤشر الصورة . بحاجة إلى مساحة كافية لحفظ الصورة. تنسخ صورة من الشاشة إلى الذاكرة . إحداثيات المستطيل تحدد منطقة الشاشة التي س وف يتم نسخها . Bitmap يشير إلى منطقة الذاكرة التي سوف يوضع فيها الصورة.
الصيغة العامة لهذه الدالة هي:

```
void getimage(int left, int top, int right, int bottom, void *bitmap);
```

41.2 دالة putimage()

تستخدم هذه الدالة لعرض (رسم) صورة على الشاشة. تضع الصورة التي تم حفظها سابقاً باستخدام الأمر getimage تسترجع الصورة على الشاشة في الزاوية العلوية اليسرى للمستطيل. Ptr يشير إلى منطقة الذاكرة التي تحتوي الصورة التي تم حفظها باستخدام getimage. المعامل OP يحدد العملية التي تتحكم في كيفية حساب لون بكسل ال وجهة على الشاشة، على أساس البكسل الموجودة بالفعل على الشاشة، وبكسل المصدر المقابلة في الذاكرة . المعامل OP يأخذ القيم COPY_PUT, XOR_PUT, OR_PUT, AND_PUT, أو NOT_PUT.

الصيغة العامة لهذه الدالة هي:

```
void putimage(int left, int top, void *ptr, int op);
```

42.2 دالة imagesize()

تستخدم هذه الدالة لحساب حجم الصورة لمستطيل مع طى (معين) . تعيد عدد البايتات اللازمة لحفظ الصورة ويستخدم مع الدالة getimage

الصيغة العامة لهذه الدالة هي:

```
unsigned int imagesize(int left, int top, int right, int bottom);
```

مثال:

لو أردنا حفظ شكل على الذاكرة وعرضه على الشاشة:

الحل:

نكتب الجمل التالية:

```
circle(200, 200, 50);
 line(150, 200, 250, 200);
 line(200, 150, 200, 250);

 int bytes = imagesize(150, 150, 250, 250);
 sprintf(array, "Number of bytes required to store required area = %d", bytes);
 outtextxy(10, 280, array);


 void far *image = 0;
   int sz = imagesize(left, top - 35, right + 35, bottom);
   image = farmalloc(sz);

  // getimage
   getimage(left, top - 35, right + 35, bottom, image);
   putimage(left, bottom + 10, image, NOT_PUT);
   putimage(left, top - 170, image, COPY_PUT);
```

تدريب 40:

اكتب برنامجاً بلغة سي يرسم مستطيل أبعاده (100,200,200,275) ثم يضع نسخة منه في الذاكرة وبعد ذلك يسترجع المستطيل من الذاكرة ويرسمه على الشاشة عند إحداثيات نقطة جديدة (100, 340).

الحل:

```
#include<stdio.h>
 #include<alloc.h>
 #include<graphics.h>
 #include<conio.h>
 int main()
 {       int gd=DETECT, gm,size;
        void far *buf=0;
        initgraph(&gd,&gm,"c:\\tc\\bgi");
        outtextxy(100,80,"Original  image:");
        rectangle(100,200,200,275);
```

```
        size=imagesize(100,200,200,275);
        buf=farmalloc(size);
        getimage(100,200,200,275,buf);
        outtextxy(100,320,"Captured  image:");
        putimage(100,340,buf,COPY_PUT);
        getch();
        closegraph();
    return 0;
}
```

43.2 دالة delay()

تستخدم هذه الدالة لوقف تنفيذ البرنامج (الإنتظار) لوقت محدد يقاس بالميلي ثانية. (1 second = 1000 milliseconds) وأن استخدام الدالة delay يتطلب تضمين البرنامج ملف الترويسة (المكتبة) dos.h.

الصيغة العامة لهذه الدالة هي:

```
void delay(unsigned int);
```

مثال:
لو أردنا أن نخرج من البرنامج بعد 10 ثواني.
الحل:
بعد أن يطبع النص البرنامج ينتظر 10000 ميلي ثانية أو 10 نواني ثم يتوقف.

```
#include<stdio.h>
#include<stdlib.h>
 main()
{
   printf("This c program will exit in 10 seconds.\n");

   delay(10000);
    return 0;
}
```

44.2 دالة kbhit()

هذه الدالة تستخدم عند الحاجة للاستمرار في تنفيذ عمل ما حتى الضغط على أحد أزرار لوحة المفاتيح. فهي تقوم بقراءة بفر لوحة المفاتيح keyboard buffer وتكون قيمته العددية صفراً حتى يتم الضغط على أحد أزرار لوحة المفاتيح فتتغير قيمته بقيمة الآسكي للزر الذي يتم ضغطه. عادةً ما نستخدم هذه الدالة مع برامج الرسم التي تحتوي على حركة ونريد الاستمرار في عرض الحركة حتى يتم الضغط على أحد أزرار لوحة المفاتيح.
هذه الدالة ليست معرفة ضمن مجموعة دوال لغة سي المعيارية، ولهذا لا تعمل مع كل المترجمات. وتعيد قيمة عددية صحيحة ليست صفراً. وهي تنتمي إلى ملف الترويسة conio.h.
الصيغة العامة لهذه الدالة هي:

```
int kbhit(void)
```

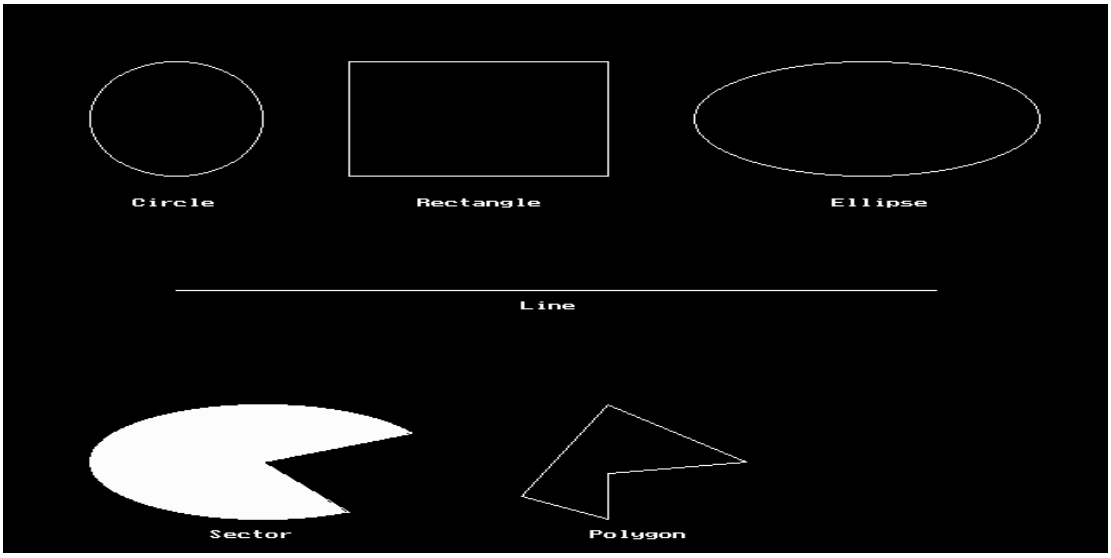
هذه الدالة ليس لها معاملات.
مثال:
لو أردنا من البرنامج أن يستمر في التنفيذ حتى يتم ضغط زر على لوحة المفاتيح
الحل:
نكتب الجملة التالية في البرنامج

```
While(!kbhit())
```

القسم الثالث:

تطبيقات على دوال الرسم

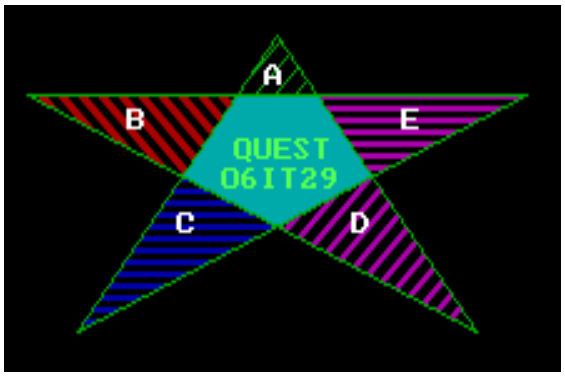
تدريب 41:
اكتب برنامجاً بلغة سي لرسم الأشكال الموضحة في الشكل التالي:



الحل:

```
#include<graphics.h>
#include<conio.h>
void main()
{   int gd=DETECT, gm;
    int poly[12]={350,450, 350,410, 430,400, 350,350, 300,430, 350,450 };

    initgraph(&gd, &gm, "c:\\tc\\BGI");
         circle(100,100,50);
    outtextxy(75,170, "Circle");
    rectangle(200,50,350,150);
    outtextxy(240, 170, "Rectangle");
    ellipse(500, 100,0,360, 100,50);
    outtextxy(480, 170, "Ellipse");
    line(100,250,540,250);
    outtextxy(300,260,"Line");
    sector(150, 400, 30, 300, 100,50);
    outtextxy(120, 460, "Sector");
    drawpoly(6, poly);
    outtextxy(340, 460, "Polygon");
    getch();
    closegraph();       }
```

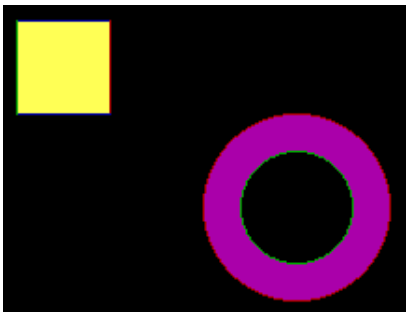
تدريب 42:
اكتب برنامجا بلغة سي لرسم الشكل التالي:



الحل:

```
#include<graphics.h>
#include<dos.h>
#include<conio.h>
void main(void)
{int gd=DETECT,gm;
initgraph(&gd,&gm,"C:\\tc\\bgi");
int intstar[]={200,200,280,100,360,200,180,120,380,120,200,200};
setcolor(2);
drawpoly(7,intstar);
setfillstyle(1,3);
floodfill(280,130,2);
setfillstyle(2,1);
floodfill(210,195,2);
setfillstyle(3,2);
floodfill(280,110,2);
setfillstyle(4,5);
floodfill(350,190,2);
setfillstyle(5,4);
floodfill(190,121,2);
setfillstyle(2,5);
floodfill(370,121,2);
setcolor(WHITE);
outtextxy(330,125,"E");
outtextxy(220,125,"B");
outtextxy(275,110,"A");
outtextxy(240,160,"C");
outtextxy(310,160,"D");
for(int i=1;i<32;i++)
{setcolor(i);
outtextxy(263,135,"QUEST");
outtextxy(258,145,"06IT29");
delay(600);}
getch();
closegraph();
}
```

تدريب 43:
اكتب برنامجاً بلغة سي لرسم الشكل التالي:



الحل:

```
#include< stdio.h>
#include< conio.h>
#include< graphics.h>
void floodFill4(int,int,int,int);
void floodFill8(int,int,int,int);
void main()
{
 int gd=DETECT,gm;
 int b=0;
 initgraph(&gd,&gm,"");

 setbkcolor(b);
 setcolor(BLUE);
 outtextxy(5,5,"Flood Filling algorithm using 4-connected and 8-connected");
 line(100,100,50,100);
 setcolor(RED);
 line(100,100,100,150);
 setcolor(BLUE);
 line(100,150,50,150);
 setcolor(GREEN);
 line(50,100,50,150);
 floodFill8(53,102,14,b);
 setcolor(GREEN);
 circle(200,200,30);
 setcolor(RED);
 circle(200,200,50);
 floodFill4(200,240,5,b);
getch();
}
void floodFill4(int x,int y,int fill,int old)
{
 int current;
 current=getpixel(x,y);
 //delay(100);
 if(current==old)
 {
  putpixel(x,y,fill);
  floodFill4(x+1,y,fill,old);
  floodFill4(x-1,y,fill,old);
  floodFill4(x,y-1,fill,old);
  floodFill4(x,y+1,fill,old);
```

```
 }
}
void floodFill8(int x,int y,int fill,int old)
{
 int current;
 current=getpixel(x,y);
 if(current==old)
 {
  putpixel(x,y,fill);
  floodFill8(x+1,y,fill,old);
  floodFill8(x-1,y,fill,old);
  floodFill8(x,y-1,fill,old);
  floodFill8(x,y+1,fill,old);
  floodFill8(x+1,y+1,fill,old);
  floodFill8(x-1,y+1,fill,old);
  floodFill8(x-1,y-1,fill,old);
  floodFill8(x+1,y-1,fill,old);
 }
}
```

تدريب 44:
اكتب برنامجاً بلغة سي لرسم الشكل الآتي ويتحرك بإتجاه اليمين.

الحل:

```
#include<conio.h>
#include<dos.h>
#include<graphics.h>

void car(int x, int c, int i_f_y, int i_r_y)
{
   setcolor(c);
   line(x + 150, 100+i_r_y, x + 242, 100+i_f_y);
   ellipse(x + 242, 105+i_f_y, 0, 90, 10, 5);
   line(x + 150, 100+i_r_y, x + 120, 150+i_r_y);
   line(x + 252, 105+i_f_y, x + 280, 150+i_f_y);
   line(x + 100, 150+i_r_y, x + 320, 150+i_f_y);
   line(x + 100, 150+i_r_y, x + 100, 200+i_r_y);
   line(x + 320, 150+i_f_y, x + 320, 200+i_f_y);
   line(x + 100, 200+i_r_y, x + 110, 200+i_r_y);
   line(x + 320, 200+i_f_y, x + 310, 200+i_f_y);
   arc(x + 130, 200+i_r_y, 0, 180, 20);
   arc(x + 290, 200+i_f_y, 0, 180, 20);
   line(x + 270, 200+i_f_y, x + 150, 200+i_r_y);
   circle(x + 130, 200+i_r_y, 17);
   circle(x + 290, 200+i_f_y, 17);
```

```
}

void main()
{
   int gd = DETECT, gm = DETECT, i = -200;
   int iy=0, prv_iy=0, prv_iry=0;
   initgraph(&gd, &gm, "");

   ellipse(318, 220, 0, 180, 15, 8);
   line(0,220,640,220);
   setfillstyle(SOLID_FILL, RED);
   floodfill(318,215,15);
   car(i, 15,0,0);

   while (!kbhit()) {
                car(i++, 0,prv_iy,prv_iry);
                if(i>0 && i<50) {
                        car(i,15,iy,0);
                        prv_iy = iy;
                        if(i>25){
                                if(i%2==0) iy++;
                        } else {
                                if(i%2==0) iy--;
                        }
                } else if(i>160 && i<210) {
                        car(i,15,0,iy);
                        prv_iry = iy;
                        if(i>185){
                                if(i%2==0) iy++;
                        } else {
                                if(i%2==0) iy--;
                        }
                } else if(i>500) {
                        break;
                } else {
                        prv_iy = 0;
                        prv_iry = 0;
                        iy = 0;
                        car(i, 15,0,0);
                }
                delay(20);
   }
}
```

تدريب 45:

اكتب برنامجاً بلغة سي لرسم الشكل الآتي ويتحرك بإتجاه اليمين.

الحل:

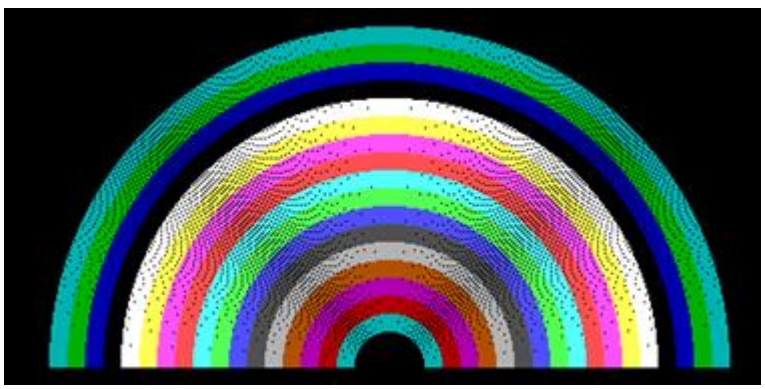
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void main()
{
int gd=DETECT,gm,i=-300,j;
int poly[16]={100,100,250,100,250,50,300,50,325,90,325,140,100,140,100,100};
int tpoly[16]={100,100,250,100,250,50,300,50,325,90,325,140,100,140,100,100};
initgraph(&gd,&gm,"");
getch();
while(!kbhit())
{
for(j=0;j<16;j+=2)
{
poly[j]=tpoly[j]+i;
}
fillpoly(8,poly);
setfillstyle(5,7);
bar(275+i,60,295+i,85);
setfillstyle(5,8);
fillellipse(140+i,140,20,20);
fillellipse(280+i,140,20,20);
setfillstyle(1,0);
fillellipse(140+i,140,10,10);
fillellipse(280+i,140,10,10);
setcolor(15);
line(0,160,639,160);
setcolor(0);
setfillstyle(1,4);
delay(20);
cleardevice();
i++;
if(i>550)
i=-300;
}
closegraph();
}
```

تدريب 46:
اكتب برنامجاً لرسم الشكل الآتي ويتحرك إلى الأعلى ثم إلى الأسفل ball bouncing.

الحل:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>

void main()
{
   int gd = DETECT, gm = DETECT;
   int x, y = 0, j, t = 400, c = 1;
   initgraph(&gd, &gm, "");
   setcolor(RED);
   setfillstyle(SOLID_FILL, RED);
   for (x = 40; x < 602; x++) {
         cleardevice();
         circle(x, y, 30);
         floodfill(x, y, RED);
         delay(40);

         if (y >= 400) {
            c = 0;
            t -= 20;
         }
         if (y <= (400 - t))
            c = 1;
         y = y + (c ? 15 : -15);
    }

   getch();
}
```

تدريب 47:
اكتب برنامجاً بلغة سي لرسم الشكل الآتي (أقواس يزداد نصف قطرها ويتغير لونها).



الحل:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void main()
{
//auto detection
      int gdriver = DETECT,gmode;
      int x,y,i;
      //initialize graphics mode(passed three arguments to initgraph function)
      initgraph(&gdriver,&gmode,"C:\\Turboc3\\BGI");/*&gdriver is the address of gdriver
variable,&gmodeis the address of gmode and

,"C:\\Turboc3\\BGI" is the path where BGI files are stored.*/
        x=getmaxx()/2;
         y=getmaxy()/2;
         for(i=30;i<200;i++)
           {
                  delay(100);
                   setcolor(i/10);
                    arc(x,y,0,180,i-10);
            }
getch();
}
```

تدريب 48:

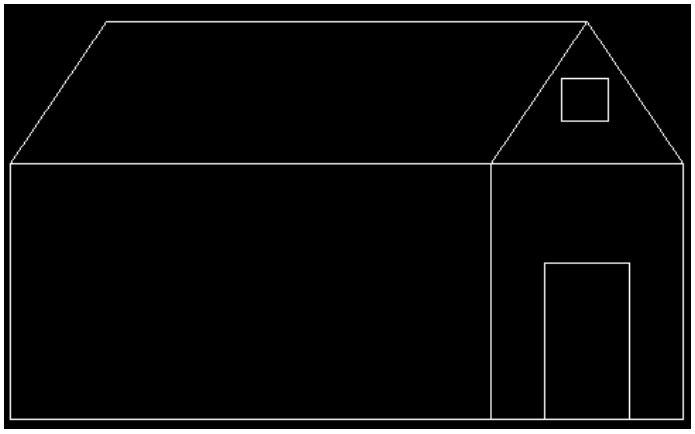
اكتب برنامجاً بلغة سي يستخدم مفهوم التعبئة الذي يدعى boundry fill. هذه الخوارزمية تبدأ من نقطة داخل الشكل وتعبئ المنطقة بلون يتم تحديده وتستمر بالتعبئة حتى تصل إلى لون حدود الشكل المختلف. أرسم الشكل الذي تريد.

الحل:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void boundary_fill(int x, int y, int fcolor, int bcolor)
{
if ((getpixel(x, y) != fcolor) && (getpixel(x, y) != bcolor)) {
putpixel(x, y, fcolor);
boundary_fill(x + 1, y, fcolor, bcolor);
boundary_fill(x - 1, y, fcolor, bcolor);
boundary_fill(x, y - 1, fcolor, bcolor);
boundary_fill(x, y + 1, fcolor, bcolor);
boundary_fill(x + 1, y - 1, fcolor, bcolor);
boundary_fill(x + 1, y + 1, fcolor, bcolor);
boundary_fill(x - 1, y - 1, fcolor, bcolor);
boundary_fill(x - 1, y + 1, fcolor, bcolor);
}
}
void main()
{
int x, y, fcolor, bcolor;
```

```
clrscr();
printf("Enter the seed point (x,y) : ");
scanf("%d%d", &x, &y);
printf("Enter boundary color : ");
scanf("%d", &bcolor);
printf("Enter new color : ");
scanf("%d", &fcolor);

int gd = DETECT, gm = DETECT;
initgraph(&gd, &gm, "");
cleardevice();
/*
please ceate your own shapes
to make a closed area.
*/
boundary_fill(x, y, fcolor, bcolor);
getch();
}
```

تدريب 49:
اكتب برنامجاً بلغة سي لرسم الشكل الآتي:

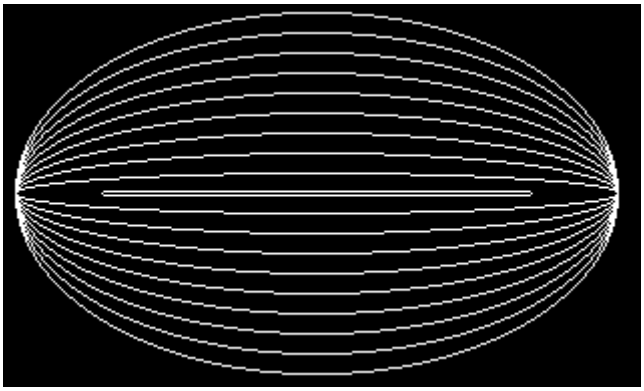


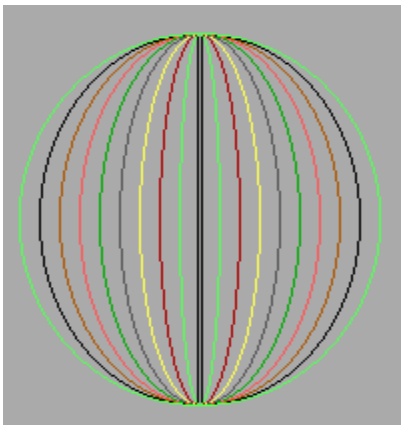
الحل:

```
#include<conio.h>
#include<stdio.h>
#include<graphics.h>
#define _num 0
void main(void)
{clrscr();int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\tc\\bgi");

line(140,80,480,80); //top horizontal
line(140,80,72,180); //left point diagonal
line(480,80,412,180); // right point diagonal -1
line(480,80,548,180); // right point diagonal -2
line(72,180,548,180); //mid Hut horizontal
line(72,180,72,360); //mid Hut horizontal to vertical down
line(412,180,412,360);//second vertical
line(548,180,548,360); //third vertical
line(72,360,548,360); // Bottom Line horizontal
line(450,250,510,250); //door of the HUT
```

```
line(450,250,450,360);
line(510,250,510,360);
line(462,120,495,120);//window
line(462,120,462,150);
line(495,120,495,150);
line(462,150,495,150);
getch();
closegraph();
}
```

تدريب 50:
اكتب برنامجاً بلغة سي لرسم الأشكال التالية على التوالي . الرسم يبدأ من الشكل الداخلي وينتقل إلى الخارجي الذي يليه حتى يكتمل الشكل:





الحل:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void main(void)
{int gd=DETECT, gm;
initgraph(&gd,&gm,"C:\\tc\\bgi");
int xE=350,yE=200; // Center of the ellipse.
int xRAD=150,yRAD;
int STANGLE=0,ENDANGLE=360;
for(yRAD=0;yRAD<100;yRAD+=10)
{ellipse(xE,yE,STANGLE,ENDANGLE,xRAD,yRAD);
delay(2000);
}//getch();
delay(2000);
clrscr();
for(xRAD=0;xRAD<100;xRAD+=10)
{setcolor(xRAD);
ellipse(xE,yE,STANGLE,ENDANGLE,xRAD,yRAD);
delay(2000);
}getch();
closegraph();
}
```

تدريب 51:

اكتب برنامجاً بلغة سي لرسم الشكل التالي واستخدم مفهوم boundry fill للتعبئة.

الحل:

```
#include< stdio.h>
#include< conio.h>
#include< graphics.h>
void BoundaryFill4(int,int,int,int);
void BoundaryFill8(int,int,int,int);
void main()
{
 int gd=DETECT,gm;
 initgraph(&gd,&gm,"");

 setbkcolor(15);
 setcolor(BLUE);
 outtextxy(0,0,"Boundary Filling algorithm using 4-connected and 8-connected");
 rectangle(50,50,100,100);
 circle(50,50,30);
 BoundaryFill8(56,56,5,BLUE);
 BoundaryFill4(40,40,14,BLUE);

getch();
}
void BoundaryFill4(int x,int y,int fill,int boundary)
{
 int current;
 current=getpixel(x,y);
 if((current!=boundary) && (current!=fill))
 {
  putpixel(x,y,fill);
  BoundaryFill4(x+1,y,fill,boundary);
  BoundaryFill4(x-1,y,fill,boundary);
  BoundaryFill4(x,y-1,fill,boundary);
  BoundaryFill4(x,y+1,fill,boundary);
 }
}
void BoundaryFill8(int x,int y,int fill,int boundary)
{
 int current;
 current=getpixel(x,y);
 if((current!=boundary) && (current!=fill))
 {

  putpixel(x,y,fill);
  BoundaryFill8(x+1,y,fill,boundary);
```

```
 BoundaryFill8(x-1,y,fill,boundary);
 BoundaryFill8(x,y-1,fill,boundary);
 BoundaryFill8(x,y+1,fill,boundary);
 BoundaryFill8(x+1,y+1,fill,boundary);
 BoundaryFill8(x-1,y+1,fill,boundary);
 BoundaryFill8(x-1,y-1,fill,boundary);
 BoundaryFill8(x+1,y-1,fill,boundary);
 }
}
```

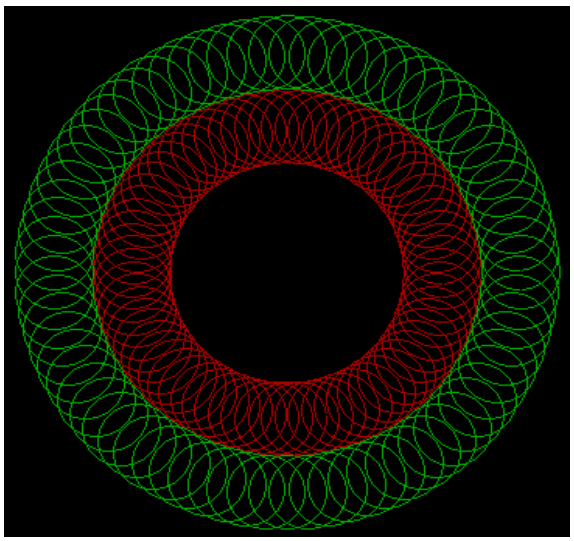
تدريب 52:
اكتب برنامجاً بلغة سي لتطبيق عمل الإشارة الضوئية:

الحل:

```
/* This program will simulate a traffic light */
#include<graphics.h>
 #include<conio.h>
 #include<dos.h>
 #include<stdlib.h>
 main()
 {
   int gd = DETECT, gm, midx, midy;
   initgraph(&gd, &gm, "..\\bgi");

   midx = getmaxx()/2;
   midy = getmaxy()/2;
   setcolor(RED);
   settextstyle(SCRIPT_FONT, HORIZ_DIR, 3);
   settextjustify(CENTER_TEXT, CENTER_TEXT);
   outtextxy(midx, midy-10, "Traffic Light Simulation");
   outtextxy(midx, midy+10, "Press any key to start");
   getch();
   cleardevice();
   setcolor(WHITE);
   settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
   rectangle(midx-30,midy-80,midx+30,midy+80);
   circle(midx, midy-50, 22);
   setfillstyle(SOLID_FILL,RED);
   floodfill(midx, midy-50,WHITE);
   setcolor(BLUE);
   outtextxy(midx,midy-50,"STOP");
   delay(1000);
   graphdefaults();
   cleardevice();
   setcolor(WHITE);
   rectangle(midx-30,midy-80,midx+30,midy+80);
   circle(midx, midy, 20);
   setfillstyle(SOLID_FILL,YELLOW);
   floodfill(midx, midy,WHITE);
   setcolor(BLUE);
   outtextxy(midx-18,midy-3,"READY");
   delay(2000);
   cleardevice();
   setcolor(WHITE);
```

```
   rectangle(midx-30,midy-80,midx+30,midy+80);
   circle(midx, midy+50, 22);
   setfillstyle(SOLID_FILL,GREEN);
   floodfill(midx, midy+50,WHITE);
   setcolor(BLUE);
   outtextxy(midx-7,midy+48,"GO");
   setcolor(RED);
   settextstyle(SCRIPT_FONT, HORIZ_DIR, 4);
   outtextxy(midx-150, midy+100, "Press any key to exit...");

   getch();
   closegraph();
   return 0;
}
```

تدريب 53:
اكتب برنامجاً بلغة سي لرسم دائرة داخل دائرة لتظهر بالشكل التالي:



الحل:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<dos.h>
main()
{
  int gd = DETECT, gm, x, y, color, angle = 0;
  struct arccoordstype a, b;

  initgraph(&gd, &gm, "C:\\TC\\BGI");
  delay(2000);


  while(angle<=360)
  {
    setcolor(BLACK);
    arc(getmaxx()/2,getmaxy()/2,angle,angle+2,100);
    setcolor(RED);
    getarccoords(&a);
    circle(a.xstart,a.ystart,25);
    setcolor(BLACK);
```

```
    arc(getmaxx()/2,getmaxy()/2,angle,angle+2,150);
    getarccoords(&a);
    setcolor(GREEN);
    circle(a.xstart,a.ystart,25);
    angle = angle+5;
    delay(50);
    }
  getch();
  closegraph();
  return 0;
  }
```

القسم الرابع

تطبيقات برمجية على بعض خوارزميات الرسم

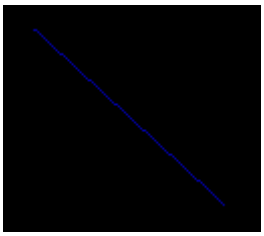
تدريب 54:
أوجد ناتج تنفيذ البرنامج التالي إذا تم إدخال البيانات التالية: خوارزمية DDA

```
Enter the value of x1 : 60
Enter the value of y1 : 56
Enter the value of x2 : 155
Enter the value of y1 : 144
```

```
 //c program for dda algorithm :
#include <dos.h>
#include <conio.h>
#include <graphics.h>
#include <stdio.h>
#include <math.h>
int main( )
{
   float x,y,x1,y1,x2,y2,dx,dy,pixel;
   int i,gd,gm;
   printf("Enter the value of x1 : ");
   scanf("%f",&x1);
   printf("Enter the value of y1 : ");
   scanf("%f",&y1);
   printf("Enter the value of x2 : ");
   scanf("%f",&x2);
   printf("Enter the value of y1 : ");
   scanf("%f",&y2);
   detectgraph(&gd,&gm);
   initgraph(&gd,&gm,"");
   dx=abs(x2-x1);
   dy=abs(y2-y1);
   if(dx>=dy)
   pixel=dx;
   else
   pixel=dy;
   dx=dx/pixel;
   dy=dy/pixel;
   x=x1;
   y=y1;
   i=1;
   while(i<=pixel)
   {
       putpixel(x,y,1);
       x=x+dx;
       y=y+dy;
       i=i+1;
       delay(100);
   }
   getch();
   closegraph();
return 0;
```

```
}
```

الحل:



تدريب 55:

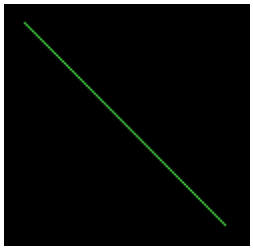
أوجد ناتج تنفيذ البرنامج التالي إذا تم إدخال البيانات التالية: خوارزمية Bresenham

```
Enter Value of X1: 200
Enter Value of Y1: 100
Enter Value of X2: 300
Enter Value of Y2: 300
```

```
//Line Drawing Algorithm - Bresenham

#include<dos.h>
#include<math.h>
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
    int gd = DETECT, gm;
    int dx, dy, p, end;
    float x1, x2, y1, y2, x, y;
    initgraph(&gd, &gm, "c:\tc\bgi");
    printf("Enter Value of X1: ");
    scanf("%f", &x1);
    printf("Enter Value of Y1: ");
    scanf("%f", &y1);
    printf("Enter Value of X2: ");
    scanf("%f", &x2);
    printf("Enter Value of Y2: ");
    scanf("%f", &y2);
    dx = abs(x1 - x2);
    dy = abs(y1 - y2);
    p = 2 * dy - dx;
    if(x1 > x2)
    {
        x = x2;
        y = y2;
        end = x1;
    }
    else
    {
        x = x1;
        y = y1;
        end = x2;
    }
```

```
    putpixel(x, y, 10);
    while(x < end)
    {
        x = x + 1;
        if(p < 0)
        {
            p = p + 2 * dy;
        }
        else
        {
            y = y + 1;
            p = p + 2 * (dy - dx);
        }
        putpixel(x, y, 10);
          delay(50);
    }
    getch();
    closegraph();
}
```

الحل:



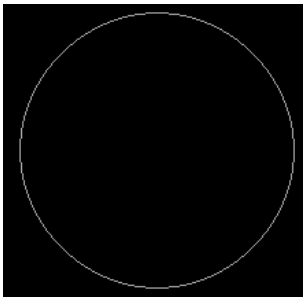
تدريب 56:
أوجد ناتج تنفيذ البرنامج التالي إذا تم إدخال البيانات التالية:

```
*** Bresenham's Circle Drawing Algorithm ***
Enter the value of Xc   200
Enter the value of yc   200
Enter the Radius of circle      100
```

```
  // Bresenham's Circle Drawing Algorithm
# include<dos.h>
# include<stdio.h>
# include<conio.h>
# include<graphics.h>
void main()
{
        int gd=DETECT,gm,xc,yc,r,x,y,Pk;
        clrscr();
        initgraph(&gd,&gm,"c:\\turboc3\\bgi");
        printf("*** Bresenham's Circle Drawing Algorithm ***\n");
        printf("Enter the value of Xc\t");
        scanf("%d",&xc);
        printf("Enter the value of yc\t");
        scanf("%d",&yc);
        printf("Enter the Radius of circle\t");
        scanf("%d",&r);
        x=0;
```

```
        y=r;
        putpixel(xc+x,yc-y,1);
        Pk=3-(2*r);
        for(x=0;x<=y;x++)
        {
                if (Pk<0)
                {
                        y=y;
                         Pk=(Pk+(4*x)+6);
                }
                else
                {
                         y=y-1;
                         Pk=Pk+((4*(x-y)+10));
                }
                putpixel(xc+x,yc-y,7);
                putpixel(xc-x,yc-y,7);
                putpixel(xc+x,yc+y,7);
                putpixel(xc-x,yc+y,7);
                putpixel(xc+y,yc-x,7);
                putpixel(xc-y,yc-x,7);
                putpixel(xc+y,yc+x,7);
                putpixel(xc-y,yc+x,7);
                delay(100);
        }
        getch();
}
```

الحل:



تدريب 57:
أوجد ناتج تنفيذ البرنامج التالي إذا تم إدخال البيانات التالية:

```
*** Ellipse Generating Algorithm ***
Enter the value of Xc   200
Enter the value of yc   200
Enter X axis length     100
Enter Y axis length     40
```

```
//  Ellipse Generating Mid-Point Algorithm
#include<dos.h>
 #include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void disp();
float x,y;
```

```
int xc,yc;
void main()
{
        int gd=DETECT,gm,a,b;
        float p1,p2;
        clrscr();
        initgraph(&gd,&gm,"c:\\turboc3\\bgi");
        printf("*** Ellipse Generating Algorithm ***\n");
        printf("Enter the value of Xc\t");
        scanf("%d",&xc);
        printf("Enter the value of yc\t");
        scanf("%d",&yc);
        printf("Enter X axis length\t");
        scanf("%d",&a);
        printf("Enter Y axis length\t");
        scanf("%d",&b);
        x=0;y=b;
        disp();
       p1=(b*b)-(a*a*b)+(a*a)/4;
        while((2.0*b*b*x)<=(2.0*a*a*y))
        {
                x++;
                if(p1<=0)
                p1=p1+(2.0*b*b*x)+(b*b);
                else
                {
                        y--;
                        p1=p1+(2.0*b*b*x)+(b*b)-(2.0*a*a*y);
                }
                disp();
                x=-x;
                disp();
                x=-x;
                delay(50);
        }
        x=a;
        y=0;
        disp();
        p2=(a*a)+2.0*(b*b*a)+(b*b)/4;
        while((2.0*b*b*x)>(2.0*a*a*y))
        {
                y++;
                if(p2>0)
                p2=p2+(a*a)-(2.0*a*a*y);
                else
                {
                        x--;
                        p2=p2+(2.0*b*b*x)-(2.0*a*a*y)+(a*a);
                }
                disp();
                y=-y;
                disp();
                y=-y;
                delay(50);
```

```
        }
        getch();
        closegraph();
}
void disp()
{
        putpixel(xc+x,yc+y,7);
        putpixel(xc-x,yc+y,7);
        putpixel(xc+x,yc-y,7);
        putpixel(xc+x,yc-y,7);
}
```

الحل:

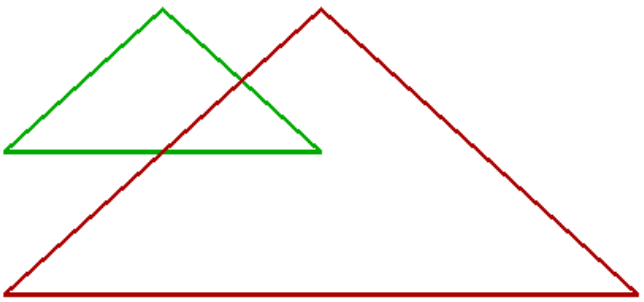
تدريب 58:
أوجد ناتج تنفيذ البرنامج التالي إذا تم إدخال البيانات التالية:

```
Number of edges: 3
Enter edge (x0,y0) : 200 10
Enter edge (x1,y1) : 100 100
Enter edge (x2,y2) : 300 100
Enter dx: 2
Enter dy: 2
Enter the center of scaling:
cx: 100
cy: 10_
```

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>


void scale( int figure[], int edges, int dx, int dy, int cx, int cy )
{
        for(int i=0; i < edges; i++)
        {
                figure[2*i] = (figure[2*i] - cx) * dx + cx;
                figure[2*i+1] = (figure[2*i+1] - cy) * dy + cy;
        }
}

void main()
{
        int figure[20], edges; // A Figure with Max 10 edges.
        int dx, dy, cx=0, cy=0;
        int gd = DETECT, gm;
        clrscr();

        printf( "Number of edges: " );
        scanf( "%d", &edges );
```

```
        for(int i=0; i < edges; i++)
        {
                printf( "Enter edge (x%d,y%d) : ", i , i );
                scanf( "%d %d", &figure[2*i], &figure[2*i+1] );
        }
        figure[2*i] = figure[0];
        figure[2*i+1] = figure[1];
        edges += 1;

        printf( "Enter dx: ");
        scanf( "%d", &dx);
        printf( "Enter dy: ");
        scanf( "%d", &dy);

        printf( "Enter the center of scaling: \n");
        printf( "cx: ");
        scanf( "%d", &cx);
        printf( "cy: ");
        scanf( "%d", &cy);

        initgraph( &gd, &gm, "" );
        cleardevice();
        setbkcolor(WHITE);

        setcolor(GREEN);
        setlinestyle(SOLID_LINE, 0, 3);
        drawpoly( edges, figure );
        getch();

        scale(figure,edges,dx,dy,cx,cy);

        setcolor(RED);
        drawpoly( edges, figure );
        getch();
}
```

الحل:



تدريب 59

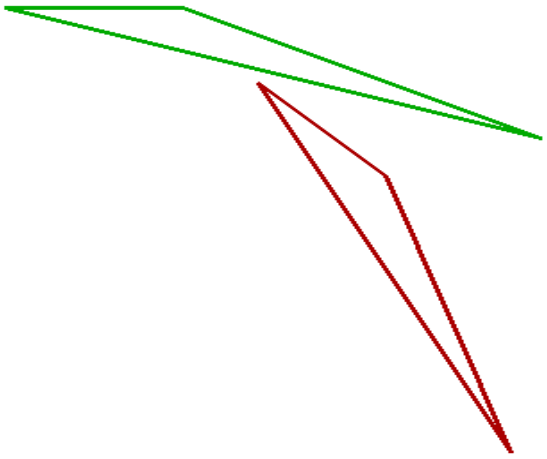
أوجد ناتج تنفيذ البرنامج التالي بعد إدخال القيم التالية:

```
Number of edges: 3
Enter edge (x0,y0) : 200 100
Enter edge (x1,y1) : 100 100
Enter edge (x2,y2) : 400 200
Enter angle of rotation in degrees: 45
Enter the center of rotation:
cx: 100 300
```

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>

void rotate( int figure[], int edges, double angle, int cx, int cy )
{
	double x, y;
	angle = -1 * (angle*3.14/180);
	double cos_a = cos(angle);
	double sin_a = sin(angle);

	for(int i=0; i < edges; i++)
	{
		x = figure[2*i] - cx;
		y = figure[2*i+1] - cy;
		figure[2*i] = ceil( (x * cos_a) - (y * sin_a) + cx );
		figure[2*i+1] = ceil( (x * sin_a)+(y * cos_a) + cy );
	}
}

void main()
{
	int figure[20], edges;  // A Figure with Max 10 edges.
	double angle;
	int cx=0, cy=0;
	int gd = DETECT, gm;
	initgraph( &gd, &gm, "" );
	int max_y = getmaxy();
	clrscr();
	cleardevice();

	printf( "Number of edges: " );
	scanf( "%d", &edges );

	for(int i=0; i < edges; i++)
	{
		printf( "Enter edge (x%d,y%d) : ", i , i );
		scanf( "%d %d", &figure[2*i], &figure[2*i+1] );
	}
	figure[2*i] = figure[0];
	figure[2*i+1] = figure[1];
	edges += 1;
```

```
        printf( "Enter angle of rotation in degrees: ");
        scanf( "%lf", &angle);

        printf( "Enter the center of rotation: \n");
        printf( "cx: ");
        scanf( "%d", &cx);
        printf( "cy: ");
        scanf( "%d", &cy);
        cy = max_y - cy;

        cleardevice();
        setbkcolor(WHITE);

        setcolor(GREEN);
        setlinestyle(SOLID_LINE, 0, 3);
        drawpoly( edges, figure );
        getch();

        for(int i=0; i < edges; i++)
                figure[2*i+1] = max_y - figure[2*i+1];
        rotate(figure,edges,angle,cx,cy);
        for(int i=0; i < edges; i++)
                figure[2*i+1] = max_y - figure[2*i+1];
        setcolor(RED);
        drawpoly( edges, figure );
        getch();
}
```

الحل:



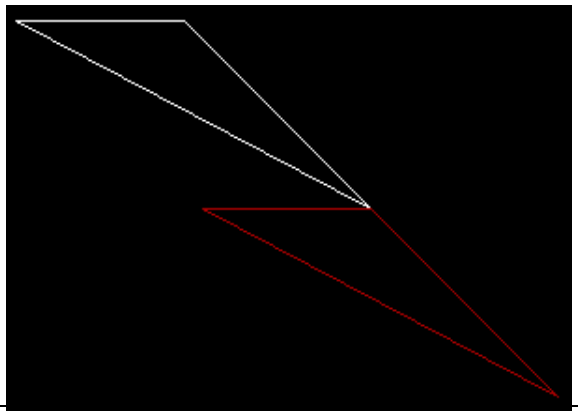
تدريب 60:
أوجد ناتج تنفيذ البرنامج التالي بعد إدخال البيانات التالية:

```
Number of edges: 3
Enter edge (x0,y0) : 100 100
Enter edge (x1,y1) : 10 100
Enter edge (x2,y2) : 200 200
Enter dx: 100
Enter dy: 100
```

```
#include<graphics.h>
```

```
#include<stdio.h>
#include<conio.h>
 void translation( int figure[], int edges, int dx, int dy )
{
        for(int i=0; i < edges; i++)
        {
                figure[2*i] += dx;
                figure[2*i+1] += dy;
        }
}
 void main()
{
        int figure[20], edges, dx, dy; // A Figure with Max 10 edges.
        int gd = DETECT, gm;
        clrscr();
        printf( "Number of edges: " );
        scanf( "%d", &edges );
        for(int i=0; i < edges; i++)
        {
                printf( "Enter edge (x%d,y%d) : ", i , i );
                scanf( "%d %d", &figure[2*i], &figure[2*i+1] );
        }
        figure[2*i] = figure[0];
        figure[2*i+1] = figure[1];
        edges += 1;
        printf( "Enter dx: ");
        scanf( "%d", &dx);
        printf( "Enter dy: ");
        scanf( "%d", &dy);

        initgraph( &gd, &gm, "" );
        cleardevice();
        drawpoly( edges, figure );
        getch();

        translation(figure,edges,dx,dy);
        setcolor(RED);
        drawpoly( edges, figure );
        getch();  }
```

الحل:



تدريب 61:

اكتب برنامجاً بلغة سي يقوم بتطبيق مفهوم الإنعكاس حول المحاور x, y origin point

الحل:

```
/*To write a program to implement 2D transformation as reflection
        ALGORITHM:
·       Declare the header file
·       Declare the variables and values
·       Using the graphics function create an image to implement the 2D reflection
·       Print the output of 2D image reflection
·       Stop the program

*/
//CODINGS:
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int a,a1,b,b1,c,c1,xt,ch;
int gd=DETECT,gm;
initgraph(&gd,&gm,"");

a=getmaxx();
a1=getmaxy();
b=a/2;
b1=a1/2;
line(b,0,b,a1);
line(0,b1,a,b1);
line(400,200,600,200);
line(400,200,400,100);
line(400,100,600,200);
printf("1.origin\n");
printf("2.x-axis\n");
printf("3.y-axis\n");
printf("4.exit\n");
do
{
printf("Enter your choice\n");
scanf("%d",&ch);
switch(ch)
{
case 1:
    c=400-b;c1=200-b1;
    line(b-c,b1-c1,b-c-200,b1-c1);
    line(b-c,b1-c1,b-c,b1-c1+100);
    line(b-c,b1-c1+100,b-c-200,b1-c1);
    break;
case 2:
    c=400-b;c1=200-b1;
    line(b+c,b1-c1,b+c+200,b1-c1);
    line(b+c,b1-c1,b+c,b1-c1+100);
    line(b+c,b1-c1+100,b+c+200,b1-c1);
    break;
case 3:
```

```
    c=400-b;c1=200-b1;
    line(b-c,b1+c1,b-c-200,b1+c1);
    line(b-c,b1+c1,b-c,b1+c1-100);
    line(b-c,b1+c1-100,b-c-200,b1+c1);
    break;
}
}while(ch<4);
getch();
closegraph();
}
//OUTPUT:
//1.    origin
//2.    x-axis
//3.    y-axis
```

تدريب 62:
اكتب برنامجاً يحقق مفهوم قصقصة الخطوط لخوارزمية كوهين سذرلاند Cohen Sutherland line clipping

الحل:

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#define MAX 20

enum { TOP = 0x1, BOTTOM = 0x2, RIGHT = 0x4, LEFT = 0x8 };

enum { FALSE, TRUE };
typedef unsigned int outcode;

outcode compute_outcode(int x, int y,
                int xmin, int ymin, int xmax, int ymax)
{
   outcode oc = 0;

   if (y > ymax)
        oc |= TOP;
   else if (y < ymin)
        oc |= BOTTOM;


   if (x > xmax)
        oc |= RIGHT;
   else if (x < xmin)
        oc |= LEFT;

   return oc;
}

void cohen_sutherland (double x1, double y1, double x2, double y2,
                double xmin, double ymin, double xmax, double ymax)
```

```
{
   int accept;
   int done;
   outcode outcode1, outcode2;

   accept = FALSE;
   done = FALSE;

   outcode1 = compute_outcode (x1, y1, xmin, ymin, xmax, ymax);
   outcode2 = compute_outcode (x2, y2, xmin, ymin, xmax, ymax);
   do
   {
         if (outcode1 == 0 && outcode2 == 0)
         {
            accept = TRUE;
            done = TRUE;
         }
         else if (outcode1 & outcode2)
         {
            done = TRUE;
         }
         else
         {
            double x, y;
            int outcode_ex = outcode1 ? outcode1 : outcode2;
            if (outcode_ex & TOP)
            {
                  x = x1 + (x2 - x1) * (ymax - y1) / (y2 - y1);
                  y = ymax;
            }

            else if (outcode_ex & BOTTOM)
            {
                  x = x1 + (x2 - x1) * (ymin - y1) / (y2 - y1);
                  y = ymin;
            }
            else if (outcode_ex & RIGHT)
            {
                  y = y1 + (y2 - y1) * (xmax - x1) / (x2 - x1);
                  x = xmax;
            }
            else
            {
                  y = y1 + (y2 - y1) * (xmin - x1) / (x2 - x1);
                  x = xmin;
            }
            if (outcode_ex == outcode1)
            {
                  x1 = x;
                  y1 = y;
                  outcode1 = compute_outcode (x1, y1, xmin, ymin, xmax, ymax);
            }
            else
            {
```

```
                x2 = x;
                y2 = y;
                outcode2 = compute_outcode (x2, y2, xmin, ymin, xmax, ymax);
            }
        }
    } while (done == FALSE);

    if (accept == TRUE)
        line (x1, y1, x2, y2);
}

 void main()
{
    int n;
    int i, j;
    int ln[MAX][4];
    int clip[4];
    int gd = DETECT, gm;

    printf ("Enter the number of lines to be clipped");
    scanf ("%d", &n);

    printf ("Enter the x- and y-coordinates of the line-endpoints:\n");
    for (i=0; i<n; i++)
        for (j=0; j<4; j++)
            scanf ("%d", &ln[i][j]);

    printf ("Enter the x- and y-coordinates of the left-top and right-");
    printf ("bottom corners\nof the clip window:\n");
    for (i=0; i<4; i++)
        scanf ("%d", &clip[i]);

    initgraph (&gd, &gm, "..//bgi");

    rectangle (clip[0], clip[1], clip[2], clip[3]);
    for (i=0; i<n; i++)
        line (ln[i][0], ln[i][1], ln[i][2], ln[i][3]);
    getch();
    cleardevice();
    rectangle (clip[0], clip[1], clip[2], clip[3]);
    for (i=0; i<n; i++)
    {
        cohen_sutherland (ln[i][0], ln[i][1], ln[i][2], ln[i][3],
            clip[0], clip[1], clip[2], clip[3]);
        getch();
    }
    closegraph();
}
```

تدريب 63:
اكتب برنامجاً بلغة سي يحقق مفهوم خوارزمية سذرلاند هجمان Sutherland-Hodgeman

الحل:

```
/*SUTHERLAND HODGEMAN POLYGON CLIPPING
The algorithm begins with an input list of all vertices in the subject polygon. Next, one side of the clip
polygon is extended infinitely in both directions, and the path of the subject polygon is traversed.
Vertices from the input list are inserted into an output list if they lie on the visible side of the extended
clip polygon line, and new vertices are added to the output list where the subject polygon path
crosses the extended clip polygon line.
*/

#include <stdio.h>
 #include <graphics.h>
 #include <conio.h>
 #include <math.h>
 #include <process.h>
 #define TRUE 1
 #define FALSE 0
 typedef unsigned int outcode;
 outcode CompOutCode(float x,float y);
 enum  {  TOP = 0x1,
 BOTTOM = 0x2,
 RIGHT = 0x4,
 LEFT = 0x8
 };
 float xmin,xmax,ymin,ymax;
 void clip(float x0,float y0,float x1,float y1)
 {
 outcode outcode0,outcode1,outcodeOut;
 int accept = FALSE,done = FALSE;
 outcode0 = CompOutCode(x0,y0);
 outcode1 = CompOutCode(x1,y1);
 do
 {
 if(!(outcode0|outcode1))
 {
 accept = TRUE;
 done = TRUE;
 }
 else
 if(outcode0 & outcode1)
 done = TRUE;
 else
 {
 float x,y;
   outcodeOut = outcode0?outcode0:outcode1;
 if(outcodeOut & TOP)
 {
 x = x0+(x1-x0)*(ymax-y0)/(y1-y0);
 y = ymax;
 }
```

```
else
if(outcodeOut & BOTTOM)
{
x = x0+(x1-x0)*(ymin-y0)/(y1-y0);
y = ymin;
}
else
if(outcodeOut & RIGHT)
{
y = y0+(y1-y0)*(xmax-x0)/(x1-x0);
x = xmax;
}
else
{
y = y0+(y1-y0)*(xmin-x0)/(x1-x0);
x = xmin;
}
if(outcodeOut==outcode0)
{
x0 = x;
y0 = y;
outcode0 = CompOutCode(x0,y0);
}
else
{
x1 = x;
y1 = y;
outcode1 = CompOutCode(x1,y1);
}
}
}while(done==FALSE);
if(accept)
line(x0,y0,x1,y1);
outtextxy(150,20,"POLYGON AFTER CLIPPING");

rectangle(xmin,ymin,xmax,ymax);
}
outcode CompOutCode(float x,float y)
{
outcode code = 0;
if(y>ymax)
code|=TOP;
else
if(y<ymin)
code|=BOTTOM;
if(x>xmax)
code|=RIGHT;
else
if(x<xmin)
code|=LEFT;
return code;
}
void main( )
{
```

```
 float x1,y1,x2,y2;
/* request auto detection */
 int gdriver = DETECT, gmode, n,poly[14],i;
 clrscr( );
 printf("Enter the no of sides of polygon:");
 scanf("%d",&n);
 printf("\nEnter the coordinates of polygon\n");
 for(i=0;i<2*n;i++)
 {
 scanf("%d",&poly[i]);
 }
 poly[2*n]=poly[0];
 poly[2*n+1]=poly[1];
 printf("Enter the rectangular coordinates of clipping window\n");
 scanf("%f%f%f%f",&xmin,&ymin,&xmax,&ymax);
/* initialize graphics and local variables */
 initgraph(&gdriver, &gmode, "c:\\tc\\bgi");

 outtextxy(150,20,"POLYGON BEFORE CLIPPING");
 drawpoly(n+1,poly);
 rectangle(xmin,ymin,xmax,ymax);
 getch( );
 cleardevice( );
 for(i=0;i<n;i++)
 clip(poly[2*i],poly[(2*i)+1],poly[(2*i)+2],poly[(2*i)+3]);
 getch( );
 restorecrtmode( );
 }

/*
OUTPUT:
Enter the no of sides of polygon:5
Enter the coordinates of polygon
50
50
200
100
350
350
80
200
40
80
Enter the rectangular coordinates of clipping window
150
150
300
300*/
```

القسم الخامس

برامج تفعيل واستخدام الفأرة للرسم

1.5 برنامج لتفعيل وإظهار مؤشر الفأرة على شاشة الرسم:

```
#include<graphics.h>
#include<conio.h>
#include<dos.h>

int initmouse();
void showmouseptr();
 union REGS i, o;
 main()
{
  int status, gd = DETECT, gm;
   initgraph(&gd,&gm,"C:\\TC\\BGI");

  status = initmouse();
   if ( status == 0 )
     printf("Mouse support not available.\n");
  else
    {  printf("Mouse support available.\n");
       showmouseptr();
     }
   getch();
  return 0;
}
```

```
int initmouse()
{
  i.x.ax = 0;
  int86(0X33,&i,&o);
  return ( o.x.ax );
}
 void showmouseptr()
{
  i.x.ax = 1;
  int86(0X33,&i,&o);

}
```

2.5 برنامج لضبط حركة مؤشر الفأرة داخل مستطيل شاشة الرسم فقط:

```
#include<dos.h>
```

```
#include<graphics.h>
#include<conio.h>
 int initmouse();
void showmouseptr();
void hidemouseptr();
void restrictmouseptr(int, int, int, int);
 union REGS i, o;
 main()
{
   int status, gd = DETECT, gm;

   initgraph(&gd,&gm,"C:\\TC\\BGI");
   settextstyle(DEFAULT_FONT,0,2);

   status = initmouse();
    if ( status == 0 )
      outtext("Mouse support not available.\n");
   else
   {
      showmouseptr();
      rectangle(120,70,520,410);
      restrictmouseptr(120,70,520,410);
   }
    getch();
   return 0;
}
 int initmouse()
{
   i.x.ax = 0;
   int86(0X33,&i,&o);
   return ( o.x.ax );
}
 void showmouseptr()
{
   i.x.ax = 1;
   int86(0X33,&i,&o);
}
 void restrictmouseptr(int x1, int y1, int x2, int y2)
{
   i.x.ax = 7;
   i.x.cx = x1;
   i.x.dx = x2;
   int86(0X33,&i,&o);

   i.x.ax = 8;
   i.x.cx = y1;
   i.x.dx = y2;
   int86(0X33,&i,&o);

}
```

3.5 برنامج لضبط حركة مؤشر الفأرة داخل دائرة على شاشة الرسم.

```
#include<graphics.h>
```

```
#include<conio.h>
#include<dos.h>
#include<stdlib.h>
#include<math.h>
 union REGS i, o;
 int initmouse()
{
   i.x.ax = 0;
   int86(0X33, &i, &o);
   return ( o.x.ax );
}
 void showmouseptr()
{
   i.x.ax = 1;
   int86(0X33, &i, &o);
}
 void hidemopuseptr()
{
   i.x.ax = 2;
   int86(0X33,&i,&o);
}
 void getmousepos(int *x, int *y)
{
   i.x.ax = 3;
   int86(0X33, &i, &o);
   *x = o.x.cx;
   *y = o.x.dx;
 }
 void movemouseptr(int x, int y)
{
   i.x.ax = 4;
   i.x.cx = x;
   i.x.dx = y;
   int86(0X33, &i, &o);
}
 main()
{
   int gd = DETECT, gm, midx, midy, radius, x, y, tempx, tempy;
    radius = 100;
    initgraph(&gd, &gm, "C:\\TC\\BGI");

   if(!initmouse())
   {
      closegraph();
      exit(1);
   }
    midx = getmaxx()/2;
   midy = getmaxy()/2;

   showmouseptr();
   movemouseptr(midx, midy);
   circle(midx, midy, radius);

   x = tempx = midx;
```

```
  y = tempy = midy;

  while(!kbhit())
  {
    getmousepos(&x, &y);

    if((pow(x-midx,2)+pow(y-midy,2)-pow(radius,2))>0)
    {
      movemouseptr(tempx, tempy);
      x = tempx;
      y = tempy;
    }

    tempx = x;
    tempy = y;
  }

  closegraph();
  return 0;}
```

4.5 برنامج يطبع رسالة مبيناً أي زر من أزرار الفأرة تم الضغط عليه:

```
#include<graphics.h>
#include<conio.h>
#include<dos.h>
 union REGS i, o;
 int initmouse()
{
  i.x.ax = 0;
  int86(0X33,&i,&o);
  return ( o.x.ax );
}
 void showmouseptr()
{
  i.x.ax = 1;
  int86(0X33,&i,&o);
}
 void getmousepos(int *button, int *x, int *y)
{
  i.x.ax = 3;
  int86(0X33,&i,&o);

  *button = o.x.bx;
  *x = o.x.cx;
  *y = o.x.dx;
}
 main()
{
  int gd = DETECT, gm, status, button, x, y;
  char array[50];

  initgraph(&gd,&gm,"C:\\TC\\BGI");
```

```
  settextstyle(DEFAULT_FONT,0,2);

  status = initmouse();

  if ( status == 0 )
    printf("Mouse support not available.\n");
  else
  {
    showmouseptr();

    getmousepos(&button,&x,&y);

    while(!kbhit())
    {
      getmousepos(&button,&x,&y);

      if( button == 1 )
      {
        button = -1;
        cleardevice();
        sprintf(array,"Left Button clicked x = %d y = %d",x,y);
        outtext(array);
      }
      else if( button == 2 )
      {
        button = -1;
        cleardevice();
        sprintf(array,"Right Button clicked x = %d y = %d",x,y);
        outtext(array);
      }

    }
  }

  getch();
  return 0;    }
```

5.5 برنامج يخفي مؤشر الفأرة من على شاشة الرسم

```
#include<graphics.h>
#include<conio.h>
#include<dos.h>
 int initmouse();
void showmouseptr();
void hidemouseptr();
 union REGS i, o;
 main()
{
  int status, count = 1, gd = DETECT, gm;
   initgraph(&gd,&gm,"C:\\TC\\BGI");

  status = initmouse();
```

```
  if ( status == 0 )
    printf("Mouse support not available.\n");
  else
  {
    showmouseptr();

    while(count<=10)
    {
      getch();
      count++;
      if(count%2==0)
        hidemouseptr();
      else
        showmouseptr();
    }
  }

  getch();
  return 0;
}

int initmouse()
{
  i.x.ax = 0;
  int86(0X33,&i,&o);
  return ( o.x.ax );
}

void showmouseptr()
{
  i.x.ax = 1;
  int86(0X33,&i,&o);
}

void hidemouseptr()
{
  i.x.ax = 2;          // to hide mouse
  int86(0X33,&i,&o);

}
```

6.5 برنامج لمعرفة إحداثيات مؤشر الفأرة (مكانه) على شاشة الرسم:

```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
```

```c
#include<dos.h>

int initmouse();
void showmouseptr();
void hidemouseptr();
void getmousepos(int*,int*,int*);

union REGS i, o;

main()
{
  int gd = DETECT, gm, status, button, x, y, tempx, tempy;
  char array[50];

  initgraph(&gd,&gm,"C:\\TC\\BGI");
  settextstyle(DEFAULT_FONT,0,2);

  status = initmouse();

  if ( status == 0 )
    printf("Mouse support not available.\n");
  else
  {
    showmouseptr();

    getmousepos(&button,&x,&y);

    tempx = x;
    tempy = y;

    while(!kbhit())
    {
      getmousepos(&button,&x,&y);

      if( x == tempx && y == tempy )
      {}
      else
      {
        cleardevice();
        sprintf(array,"X = %d, Y = %d",x,y);
        outtext(array);
        tempx = x;
        tempy = y;
      }
    }
  }

  getch();
  return 0;
}

int initmouse()
{
  i.x.ax = 0;
```

```
  int86(0X33,&i,&o);
  return ( o.x.ax );
}

void showmouseptr()
{
  i.x.ax = 1;
  int86(0X33,&i,&o);
}

void getmousepos(int *button, int *x, int *y)
{
  i.x.ax = 3;
  int86(0X33,&i,&o);

  *button = o.x.bx;
  *x = o.x.cx;
  *y = o.x.dx;    }
```

7.5 برنامج لوضع مؤشر الفأرة في مكان محدد على شاشة الرسم:

```
#include <dos.h>
#include <graphics.h>
union REGS in, out;
void set()
{
        in.x.ax=4;
        in.x.cx=150;
        in.x.dx=100;
        int86(0x33,&in,&out);     }
```

8.5 برنامج يستخدم مؤشر الفأرة بدلاً من لوحة المفاتيح لرسم شكل حر:

```
#include<conio.h>
#include <graphics.h>
#include <dos.h>
union REGS i,o;

void show_mouse()
{
        i.x.ax=1;
        int86(0x33,&i,&o);
}

void hide_mouse()
{
        i.x.ax=2;
        int86(0x33,&i,&o);
}
```

```
void get_mouse_pos(int *x,int *y,int *button)
{
        i.x.ax=3;
        int86(0x33,&i,&o);
        *x=o.x.cx;
        *y=o.x.dx;
        *button=o.x.bx&1;
}
void main()
{
        int gdriver = DETECT, gmode, errorcode,button,x1,y1,x2,y2;
        initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
//      detect_mouse ();
        outtextxy(230,400,"Press any key to exit....");
        while(!kbhit())
        {
                show_mouse();
                get_mouse_pos(&x1,&y1,&button);
                x2=x1;
                y2=y1;
                while(button==1)
                {
                        hide_mouse();
                        line(x1,y1,x2,y2) ;
                        x1=x2;
                        y1=y2;
                        get_mouse_pos(&x2,&y2,&button);
                }
        }
}
```

9.5 برنامج يستخدم مؤشر الفأرة كفرشاة للرسم كما في برنامج الرسام:

```
#include<dos.h>
#include<conio.h
#include<stdio.h>
#include<graphics.h>
#include<stdlib.h>
void main()
{
int x,y,b,px,py,c,p,s,cl;
int d=0,m;
union REGS i,o;
initgraph(&d,&m,"c:\tc");
i.x.ax=1;
int86(0x33,&i,&o);
i.x.ax=8;
i.x.cx=20;
i.x.dx=450;

int86(0x33,&i,&o);
```

```
printf("Brush style insert number from 0 to 5  : ");
scanf("%d",&p);
printf("Brush size insert number from 1  to 7  : ");
scanf("%d",&s);
printf("Brush color insert number from 1 to 16 : ");
scanf("%d",&cl);
clrscr();
cleardevice();
printf("\t\t**********DRAW IMAGE**********");
while(!kbhit())
{
i.x.ax=3;
b=o.x.bx;
x=o.x.cx;
y=o.x.dx;
px=x;
py=y;
int86(0x33,&i,&o);
if(cl==16)
{
c=random(16);
}
else
{
c=cl;
}
setcolor(c);
if(b==1)
{
i.x.ax=3;
int86(0x33,&i,&o);
x=o.x.cx;
y=o.x.dx;
b=o.x.bx;
switch(p)
{
case 1:circle(px,py,s);break;
case 2:ellipse(px,py,0,270,s,s+2);break;
case 3:fillellipse(px,py,s+2,s);break;
case 4:rectangle(px,py,x,y);break;
case 5:sector(px,py,30,120,s,s);break;
default:line(px,py,x,y);
}
}
}
getch();
restorecrtmode();
closegraph();
}
```

10.5 برنامج يرسم خطوط باستخدام مؤشر الفلوة بالنقر والجر من نقطة إلى أخرى

```
#include <graphics.h>
 #include <conio.h>
 #include <stdio.h>
```

```
#include <stdlib.h>
#include <alloc.h>
#include <dos.h>

void restore(int,int);
void get_mouse_pos(int*,int*,int*);
void save(int,int,int,int);
void show_mouse();
void hide_mouse();
 union REGS i,o;
//    char far *p;
        void far *p=0;
   void  main()
    {
        int gd=DETECT,gm,button,x1,y1,x2,y2,prevx2,prevy2,x,y;
        initgraph(&gd,&gm,"c:\\tc\\bgi");
        i.x.ax=0;
        int86(0x33,&i,&o);
        if(o.x.ax==0)
          {
            printf("No Mouse is available..");
            exit(1);
            restorecrtmode();
          }
        while(!kbhit())
        {
         show_mouse();
         get_mouse_pos(&x1,&y1,&button);

         if(button==1)
         {
          hide_mouse();
          x2=x1;
          y2=y1;
          save(x1,y1,x2,y2);
          line(x1,y1,x2,y2);
         prevx2=x2;
         prevy2=y2;
         get_mouse_pos(&x2,&y2,&button);

         while(button==1)
            {
          if(x2!=prevx2 || y2!=prevy2)
            {
                setcolor(BLACK);
                line(x1,y1,prevx2,prevy2);
                x=x1<prevx2?x1:prevx2;
                y=y1<prevy2?y1:prevy2;
                restore(x,y);
                setcolor(WHITE);
                save(x1,y1,x2,y2);
                line(x1,y1,x2,y2);
                prevx2=x2;
                prevy2=y2;
```

```
          }
          get_mouse_pos(&x2,&y2,&button);
            }
         farfree(p);
    }
    }
        restorecrtmode();
    }

void show_mouse()
    {
    i.x.ax=1;
    int86(0x33,&i,&o);
    }

void   hide_mouse()
  {
       i.x.ax=2;
         int86(0x33,&i,&o);
  }

void get_mouse_pos(int *x,int *y,int *button)
  {
        i.x.ax=3;
        int86(0x33,&i,&o);

        *x=o.x.cx;
        *y=o.x.dx;
        *button=o.x.bx&1;
       }

 void save(int x1,int y1,int x2,int y2)
  {
        unsigned area;

        area=imagesize(x1,y1,x2,y2);
        p= farmalloc(area);

        if(p==NULL)
        {
          restorecrtmode();
          printf("No Memory...");
         exit(1);
        }

       getimage(x1,y1,x2,y2,p);
    }

 void restore(int x1,int y1)
    {
  putimage(x1,y1,p,OR_PUT);
  farfree(p);
    }
```

```
//When drawing lines interactively,we must make sure that the currently
// drawn line doesn't wipe off already drawn lines when it intersects
```

11.5 برنامج تفعيل الماوس على شاشة الكتابة العادية

```
#include<dos.h>
#include<conio.h>

int initmouse();
void showmouseptr();

union REGS i, o;

main()
{
  int status;

  status = initmouse();

  if ( status == 0 )
    printf("Mouse support not available.\n");
  else
    showmouseptr();

  getch();
  return 0;
}

int initmouse()
{
  i.x.ax = 0;
  int86(0X33,&i,&o);
  return ( o.x.ax );
}
 void showmouseptr()
{
  i.x.ax = 1;
  int86(0X33,&i,&o);
}
```

12.5 برنامج لعبة الضغط على زر بحيث كلما اقترب مؤشر الفأرة من الزر يتم نقل الزر من مكانه، وهذا يتطلب معرفة إحداثيات الحالية للفأرة عند كل حركة.

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
```

```c
#include <graphics.h>
#include <stdlib.h>

union REGS i, o;
int left = 265, top = 250;

void initialize_graphics_mode()
{
 int gd = DETECT, gm, error;

 initgraph(&gd,&gm,"C:\\TC\\BGI");

 error = graphresult();

 if (error != grOk)
 {
  perror("Error ");
  printf("Press any key to exit...\n");
  getch();
  exit(EXIT_FAILURE);
 }
}

void showmouseptr()
{
 i.x.ax = 1;
 int86(0x33,&i,&o);
}

void hidemouseptr()
{
 i.x.ax = 2;
 int86(0x33,&i,&o);
}

void getmousepos(int *x,int *y)
{
 i.x.ax = 3;
 int86(0x33,&i,&o);

 *x = o.x.cx;
 *y = o.x.dx;
}

void draw_bar()
{
 hidemouseptr();
 setfillstyle(SOLID_FILL,CYAN);
 bar(190,180,450,350);
 showmouseptr();
}

void draw_button(int x, int y)
{
```

```
 hidemouseptr();
 setfillstyle(SOLID_FILL,MAGENTA);
 bar(x,y,x+100,y+30);
 moveto(x+5,y);
 setcolor(YELLOW);
 outtext("Press me");
 showmouseptr();
}

void draw()
{
 settextstyle(SANS_SERIF_FONT,HORIZ_DIR,2);
 outtextxy(155,451,"<a
href="http://www.programmingsimplified.com"">www.programmingsimplified.com"</a>);
 setcolor(BLUE);
 rectangle(0,0,639,450);
 setcolor(RED);
 outtextxy(160,25,"Try to press the \"Press me\" button");
 outtextxy(210,50,"Press escape key to exit");
 setfillstyle(XHATCH_FILL,GREEN);
 setcolor(BLUE);
 bar(1,1,75,449);
 bar(565,1,638,449);
 showmouseptr();
 draw_bar();
 draw_button(left,top);
}

void initialize()
{
 initialize_graphics_mode();

 if( !initmouse() )
 {
  closegraph();
  printf("Unable to initialize the mouse");
  printf("Press any key to exit...\n");
  getch();
  exit(EXIT_SUCCESS);
 }

 draw();
}

int initmouse()
{
 i.x.ax = 0;
 int86(0x33,&i,&o);
 return ( o.x.ax );
}

void get_input()
{
 int x, y;
```

```
while(1)
{
 getmousepos(&x,&y);

 /* mouse pointer in left of button */

 if( x >= (left-3) && y >= (top-3) && y <= (top+30+3) && x < left )
 {
  draw_bar();
  left = left + 4;

  if (left > 350)
   left = 190;

  draw_button(left,top);
 }

 /* mouse pointer in right of button */

 else if (x<=(left+100+3)&&y>=(top-3)&&y<=(top+30+3)&&x>(left+100))
 {
  draw_bar();
  left = left - 4;

  if (left < 190)
   left = 350;

  draw_button(left,top);
 }

 /* mouse pointer above button */

 else if(x>(left-3) && y>=(top-3) && y<(top) && x<= (left+100+3))
 {
  draw_bar();
  top = top + 4;

  if (top > 320)
   top = 180;

  draw_button(left,top);
 }

 /* mouse pointer below button */

 else if (x>(left-3)&&y>(top+30)&&y<=(top+30+3)&&x<=(left+100+3))
 {
  draw_bar();
  top = top - 4;

  if (top < 180)
   top = 320;
```

```
    draw_button(left,top);
   }

   if (kbhit())
   {
    if (getkey() == 1)
     exit(EXIT_SUCCESS);
   }
  }
}

int getkey()
{
 i.h.ah = 0;
 int86(22,&i,&o);

 return( o.h.ah );
}

main()
{
 initialize();

 get_input();
 return 0;
}
```

13.5 برنامج دهان بلغة سي هذه البرنامج يرسم أشكال مختلفة باستخدام الفأرة مثل خط مستقيم، دائرة، نقطة،.....الخ. وتغير اللون ومسح الشاشة. ولفهم البرنامج تابع الناتج له.

```
#include<graphics.h>
#include<dos.h>
#include<math.h>
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>

union REGS i, o;
int leftcolor[15];

int get_key()
{
  union REGS i,o;

  i.h.ah = 0;
  int86(22,&i,&o);

  return ( o.h.ah );
}

void draw_color_panel()
{
```

```
  int left, top, c, color;

  left = 100;
  top = 436;

  color = getcolor();
  setcolor(GREEN);
  rectangle(4,431,635,457);
  setcolor(RED);
  settextstyle(TRIPLEX_FONT,0,2);
  outtextxy(10,431,"Colors : ");

  for( c = 1 ; c <= 15 ; c++ )
  {
    setfillstyle(SOLID_FILL, c);
    bar(left, top, left+16, top+16);
    leftcolor[c-1] = left;
    left += 26;
  }

  setcolor(color);
}

void draw_shape_panel()
{
  int left, top, c, color;

  left = 529;
  top = 45;

  color = getcolor();
  setcolor(GREEN);
  rectangle(525,40,633,255);

  for( c = 1 ; c <= 7 ; c++ )
  {
    rectangle(left, top, left+100, top+25);
    top += 30;
  }
  setcolor(RED);
  outtextxy(530,45,"Bar");
  outtextxy(530,75,"Line");
  outtextxy(530,105,"Pixel");
  outtextxy(530,135,"Ellipse");
  outtextxy(530,165,"Freehand");
  outtextxy(530,195,"Rectangle");
  outtextxy(530,225,"Clear");
  setcolor(color);
}

void change_color(int x, int y)
{
  int c;
```

```
  for( c = 0 ; c <= 13 ; c++ )
  {
    if( x > leftcolor[c] && x < leftcolor[c+1] && y > 437 && y < 453 )
      setcolor(c+1);
    if( x > leftcolor[14] && x < 505 && y > 437 && y < 453 )
      setcolor(WHITE);
  }
}

char change_shape(int x, int y)
{
  if ( x > 529 && x < 625 && y > 45 && y < 70 )
    return 'b';
  else if ( x > 529 && x < 625 && y > 75 && y < 100 )
    return 'l';
  else if ( x > 529 && x < 625 && y > 105 && y < 130 )
    return 'p';
  else if ( x > 529 && x < 625 && y > 135 && y < 160 )
    return 'e';
  else if ( x > 529 && x < 625 && y > 165 && y < 190 )
    return 'f';
  else if ( x > 529 && x < 625 && y > 195 && y < 220 )
    return 'r';
  else if ( x > 529 && x < 625 && y > 225 && y < 250 )
    return 'c';
}

void showmouseptr()
{
  i.x.ax = 1;
  int86(0x33,&i,&o);
}

void hidemouseptr()
{
  i.x.ax = 2;
  int86(0x33,&i,&o);
}

void restrictmouseptr( int x1, int y1, int x2, int y2)
{
  i.x.ax = 7;
  i.x.cx = x1;
  i.x.dx = x2;
  int86(0x33,&i,&o);

  i.x.ax = 8;
  i.x.cx = y1;
  i.x.dx = y2;
  int86(0x33,&i,&o);
}

void getmousepos(int *button,int *x,int *y)
{
```

```
  i.x.ax = 3;
  int86(0x33,&i,&o);

  *button = o.x.bx;
  *x = o.x.cx;
  *y = o.x.dx;
}

main()
{
  int gd = DETECT,gm;

  int maxx,maxy,x,y,button,prevx,prevy,temp1,temp2,key,color;
  char ch = 'f' ;          // default free-hand drawing

  initgraph(&gd,&gm,"C:\\TC\\BGI");

  maxx = getmaxx();
  maxy = getmaxy();

  setcolor(BLUE);
  rectangle(0,0,maxx,maxy);

  setcolor(WHITE);
  settextstyle(SANS_SERIF_FONT,HORIZ_DIR,2);
  outtextxy(maxx/2-180,maxy-28,"<a
href="http://www.programmingsimplified.com"">www.programmingsimplified.com"</a>);

  draw_color_panel();
  draw_shape_panel();

  setviewport(1,1,maxx-1,maxy-1,1);

  restrictmouseptr(1,1,maxx-1,maxy-1);
  showmouseptr();
  rectangle(2,2,518,427);
  setviewport(1,1,519,428,1);

  while(1)
  {
    if(kbhit())
    {
      key = get_key();

      if( key == 1 )
      {
        closegraph();
        exit(0);
      }
    }

    getmousepos(&button,&x,&y);

    if( button == 1 )
```

```
{
  if( x > 4 && x < 635 && y > 431 && y < 457 )
    change_color( x, y );
  else if ( x > 529 && x < 625 && y > 40 && y < 250 )
     ch = change_shape( x, y );

  temp1 = x ;
  temp2 = y ;

  if ( ch == 'f' )
  {
    hidemouseptr();
    while( button == 1 )
    {
      line(temp1,temp2,x,y);
      temp1 = x;
      temp2 = y;
      getmousepos(&button,&x,&y);
    }
    showmouseptr();
  }

  while( button == 1)
    getmousepos(&button,&x,&y);

  /* to avoid interference of mouse while drawing */
  hidemouseptr();

  if( ch == 'p')
    putpixel(x,y,getcolor());

  else if ( ch == 'b' )
  {
    setfillstyle(SOLID_FILL,getcolor());
    bar(temp1,temp2,x,y);
  }
  else if ( ch == 'l')
    line(temp1,temp2,x,y);
  else if ( ch == 'e')
    ellipse(temp1,temp2,0,360,abs(x-temp1),abs(y-temp2));
  else if ( ch == 'r' )
    rectangle(temp1,temp2,x,y);
  else if ( ch == 'c' )
  {
    ch = 'f';         // setting to freehand drawing
    clearviewport();
    color = getcolor();
    setcolor(WHITE);
    rectangle(2,2,518,427);
    setcolor(color);
  }

  showmouseptr();
}
```

```
  }
}
```

تم بحمد الله

والله ولي التوفيق

أسأل الله لكم الفائدة