

B. Sc. (Information Technology)		Semester – IV	
Course Name: Computer Graphics and Animation		Course Code: USIT4P5	
Periods per week 1 Period is 50 minutes	Lectures per week	3	
		Hours	Marks
Evaluation System	Practical Examination	2½	50

1. Solve the following:

- a. Study and enlist the basic functions used for graphics in C / C++ / Python language. Give an example for each of them.

Ans:-

initgraph setcolor closegraph outtextxy putpixel delay	imagesize settextstyle resetendpt setfillstyle getpixel setlinestyle boudryfill
---	---

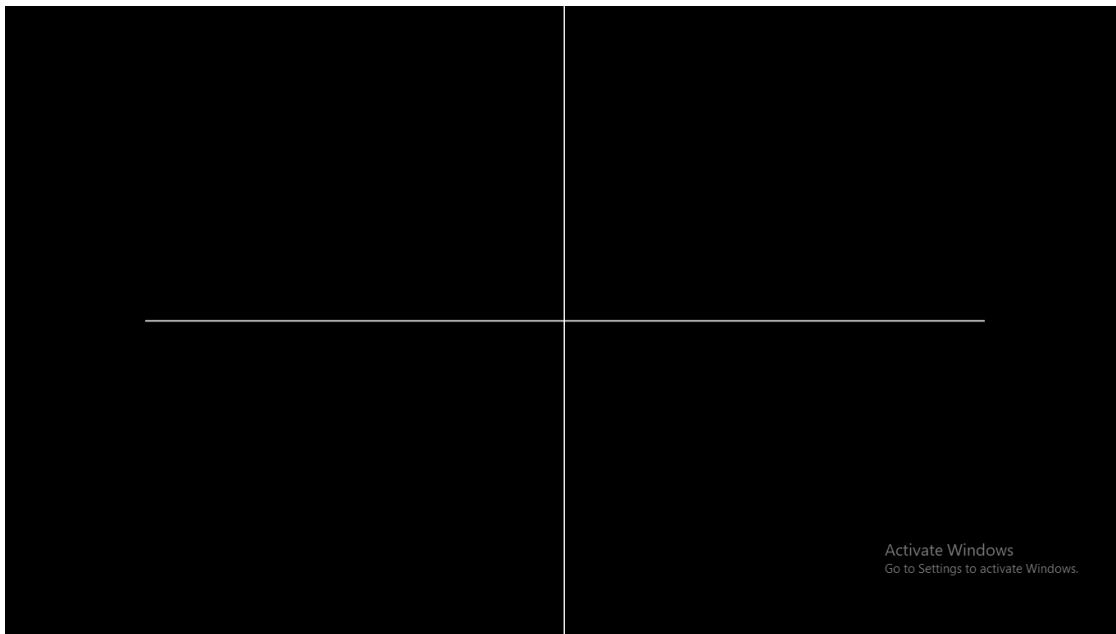
- b. Draw a co-ordinate axis at the center of the screen.

Ans:-

```

File Edit Search Run Compile Debug Project Options Window Help
SYIT_P1A.C
#include<graphics.h>
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
int main(void)
{
    int gdriver = DETECT, gmode;
    int xmax, ymax;
    initgraph(&gdriver, &gmode, "c:\\turbo3\\hg1");
    setcolor(getmaxcolor());
    xmax = getmaxx();
    ymax = getmaxy();
    line(xmax/2, 0, xmax/2, ymax);
    line(0, ymax/2, xmax, ymax/2);
    getch();
    closegraph();
    return 0;
}
9:50
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows.

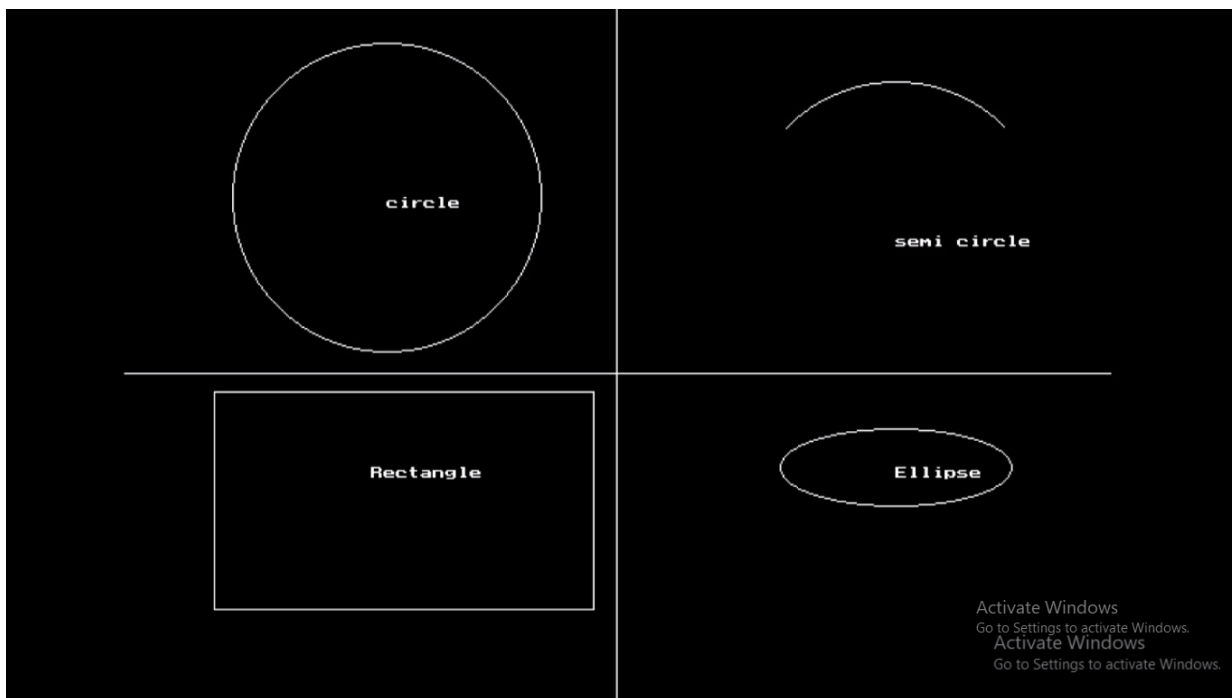
```



2. Solve the following:

- Divide your screen into four regions, draw circle, rectangle, ellipse and half ellipse in each region with appropriate message.

```
SYIT_PRA.C
#include<graphics.h>
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
int main(void)
{
    int gdriver = DETECT, gmode;
    int xmax, ymax;
    initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");
    setcolor(getmaxcolor());
    xmax = getmaxx();
    ymax = getmaxy();
    line(xmax/2,0,xmax/2,ymax);
    line(0,ymax/2,xmax,ymax/2);
    circle(170,125,100);
        outtextxy(170,125,"circle");
    rectangle(58,251,304,392);
        outtextxy(160,300, "Rectangle");
    arc(500,150,45,135,100);
        outtextxy(500,150, "semi circle");
    ellipse(500,300,0,360,75,25);
        outtextxy(500,300, "Ellipse");
    getch();
    closegraph();
    return 0;
}
```

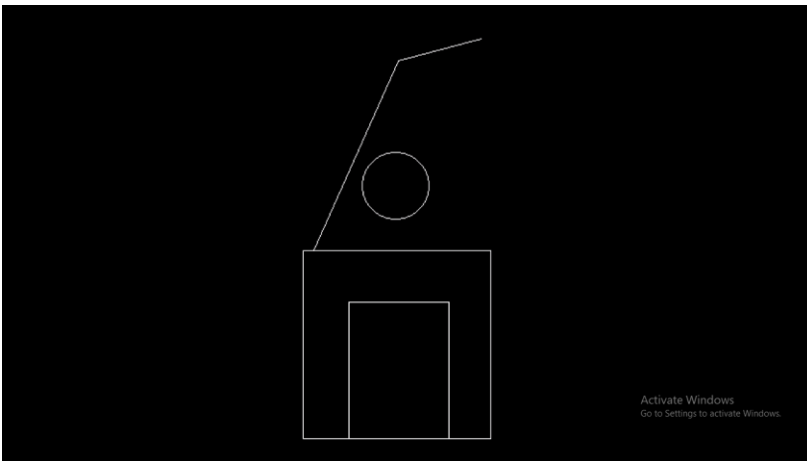


b. Draw a simple hut on the screen.

```

File Edit Search Run Compile Debug Project Options Window Help
SYIT_PRA.C 2=1+1
#include<graphics.h>
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
int main(void)
{
    int gdriver = DETECT, gmode;
    initgraph(&gdriver, &gmode, "c:\\turbo3\\bgi");
    setcolor(getmaxcolor());
    rectangle(209,257,406,454);
    rectangle(257,311,362,454);
    line(309,58,220,257);
    line(309,58,396,35);
    circle(306,189,35);
    getch();
    closegraph();
    return 0;
}
1:11
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```



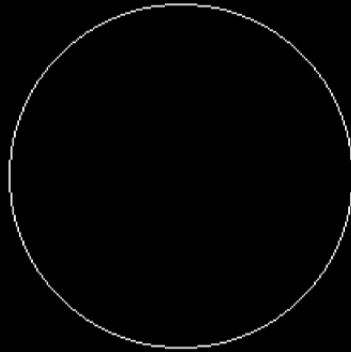
3. Draw the following basic shapes in the center of the screen :

- i. Circle
- ii. Rectangle
- iii. Square
- iv. Concentric Circles
- v. Ellipse
- vi. Line

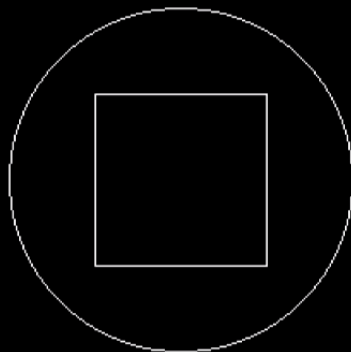
```

File Edit Search Run Compile Debug Project Options Window Help
SYIT_PRA.C 6=1+1
#include<graphics.h>
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
int main(void)
{
    int gdrive = DETECT,gmode,errorcode;
    int midx, midy, left,top,right,bottom;
    int radius=100;
    initgraph(&gdriver, &gmode, "c:\\turbo3\\bgi");
    printf("%d", getmaxx());
    midx = getmaxx() / 2;
    midy = getmaxy() / 2;
    setcolor(getmaxcolor());
    circle(midx, midy, radius);
    getch();
    left = getmaxx() /2 - 50;
    top = getmaxy() /2 - 50;
    right = getmaxx() /2 + 50;
    bottom getmaxy()/ 2 +50;
    rectangle(left,top,right,bottom);
    getch();
    rectangle(midx -20, midy -20, midx+20, midy+20);
    getch();
    ellipse(midx,midy,0,360,100,50);
    getch();
    line(0,0,midx * 2, midy * 2);
    getch();
    closegraph();
    return 0;
}

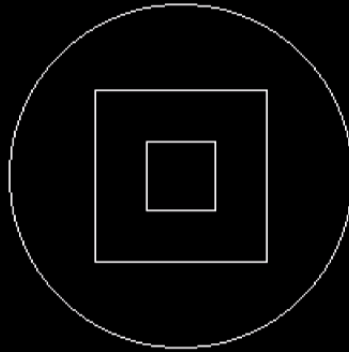
```



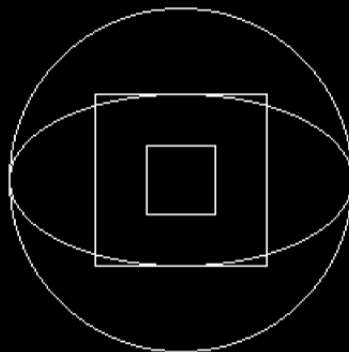
Activate Windows
Go to Settings to activate Windows.



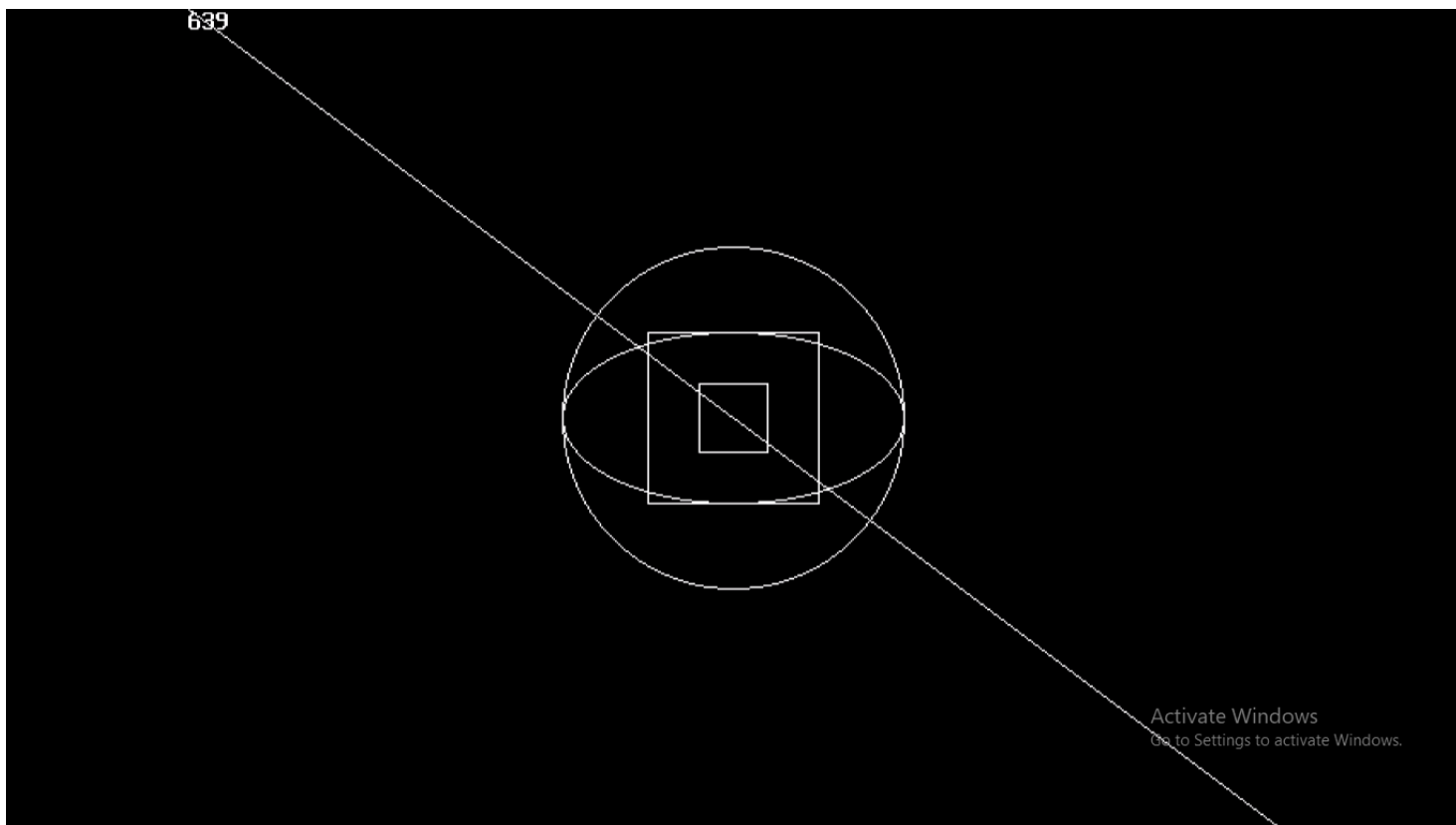
Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.



4. Solve the following:

- Develop the program for DDA Line drawing algorithm.

```

File Edit Search Run Compile Debug Project Options Window Help
[ ] SYITDDA.CPP 9=1+1
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>
#include<graphics.h>
#include<dos.h>
void ddaline(int x1, int y1, int x2, int y2);
void main(void)
{
    int x1,x2,y1,y2;
    int gd = DETECT, gm =1;
    initgraph(&gd, &gm, "c:\\\\turbo3\\\\bgi");
    cout<<"Enter the first point \n";
    cin>>x1>>y1;
    cout<<"enter the ending point value \n";
    cin>>x2>>y2;
    line(0,240,640,240);
    line(320,0,320,480);
    outtextxy(590,250, "X-axis");
    outtextxy(300,0, "Y-axis");
    ddaline(x1,y1,x2,y2);
}

```

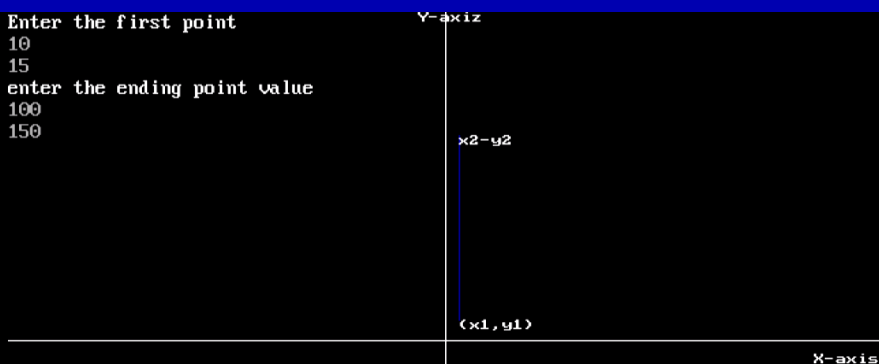
```

    getch();
    closegraph();
}
void ddaline(int x1, int y1, int x2, int y2)
{
    int dx,dy,step,k;
    float xinc,yinc,x,y;
    dx = x2-x1;
    dy = y2-y1;
    if (abs(dx)>=abs(dy))
        step = abs(dx);
    else
        step = abs(dy);
    xinc = dx/step;
    yinc = dy/step;
    x = x1;
    y = y1;
    putpixel(ceil(x) + 320, 240 - ceil(y), 1);
    outtextxy(320+x,240-y, "(x1,y1)");
    for(k=1; k<=step; k++)
    {
        xinc = dx/step;
        yinc = dy/step;
        x = x1;
        y = y1;
        putpixel(ceil(x) + 320, 240 - ceil(y), 1);
        outtextxy(320+x,240-y, "(x1,y1)");
        for(k=1; k<=step; k++)
        {
            x = x+xinc;
            y = y+yinc;
            delay(100);
            putpixel(320 + ceil(x), 240 - ceil(y), 1);
        }
        outtextxy(320 + x, 240 - y, "x2-y2");
    }
}

```

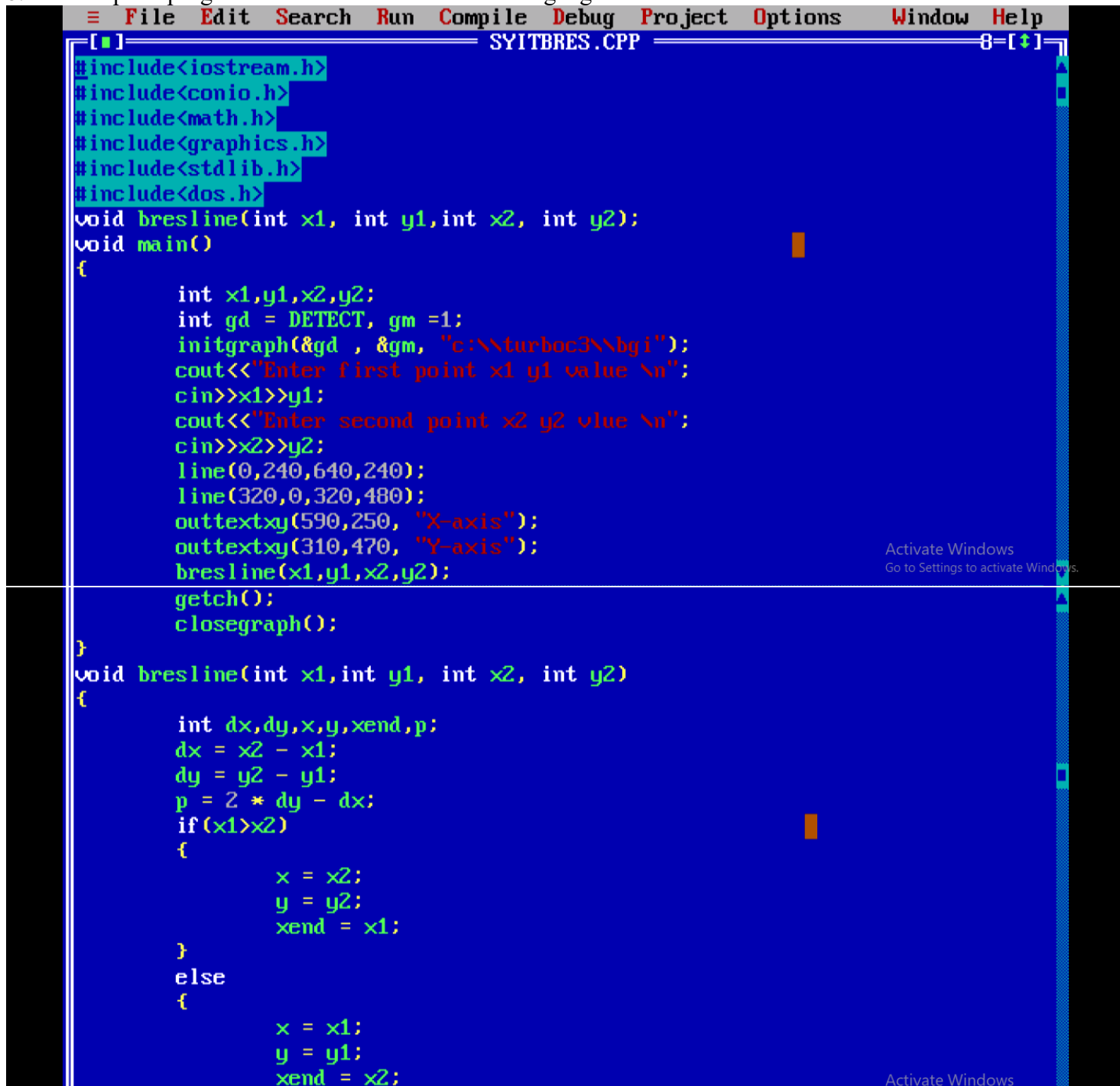
Activate Windows
Go to Settings to activate Windows.

Enter the first point
10
15
enter the ending point value
100
150



Activate Windows
Go to Settings to activate Windows.

b. Develop the program for Bresenham's Line drawing algorithm.



```
SYITBRES.CPP
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<graphics.h>
#include<stdlib.h>
#include<dos.h>
void bresline(int x1, int y1, int x2, int y2);
void main()
{
    int x1,y1,x2,y2;
    int gd = DETECT, gm =1;
    initgraph(&gd , &gm, "c:\\N\\turbo3\\bgi");
    cout<<"Enter first point x1 y1 value \n";
    cin>>x1>>y1;
    cout<<"Enter second point x2 y2 value \n";
    cin>>x2>>y2;
    line(0,240,640,240);
    line(320,0,320,480);
    outtextxy(590,250, "X-axis");
    outtextxy(310,470, "Y-axis");
    bresline(x1,y1,x2,y2);

    getch();
    closegraph();
}
void bresline(int x1, int y1, int x2, int y2)
{
    int dx,dy,x,y,xend,p;
    dx = x2 - x1;
    dy = y2 - y1;
    p = 2 * dy - dx;
    if (x1>x2)
    {
        x = x2;
        y = y2;
        xend = x1;
    }
    else
    {
        x = x1;
        y = y1;
        xend = x2;
    }
}
```

```

    }
    putpixel(ceil(x) + 320, 240 - ceil(y), 1);
    outtextxy(320+x, 240-y, "(x1,y1)");
    while(x<xend)
    {
        x = x + 1;
        if(p<0)
        {
            p = p + 2 * dy;
        }
        else
        {
            y = y + 1;
            p = p + 2 * (dy - dx);
        }
        delay(100);
        putpixel(ceil(x) + 320, 240 - ceil(y), 1);
    }
    outtextxy(320 + x, 240 - y, "(x2,y2)");
}

```

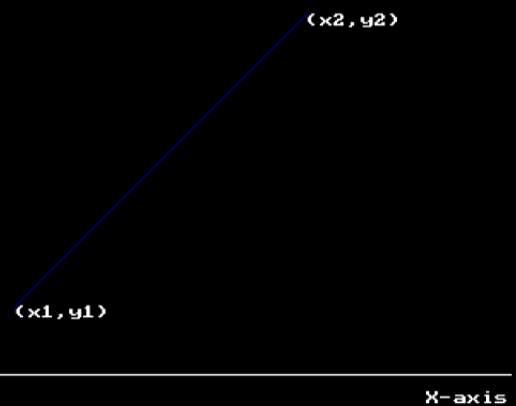
Activate Windows
Go to Settings to activate Windows.

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```

Enter first point x1 y1 value
30
40
Enter second point x2 y2 value
200
250

```



Activate Windows
Go to Settings to activate Windows.

Y-axis

5. Solve the following:

- a. Develop the program for the mid-point circle drawing algorithm.

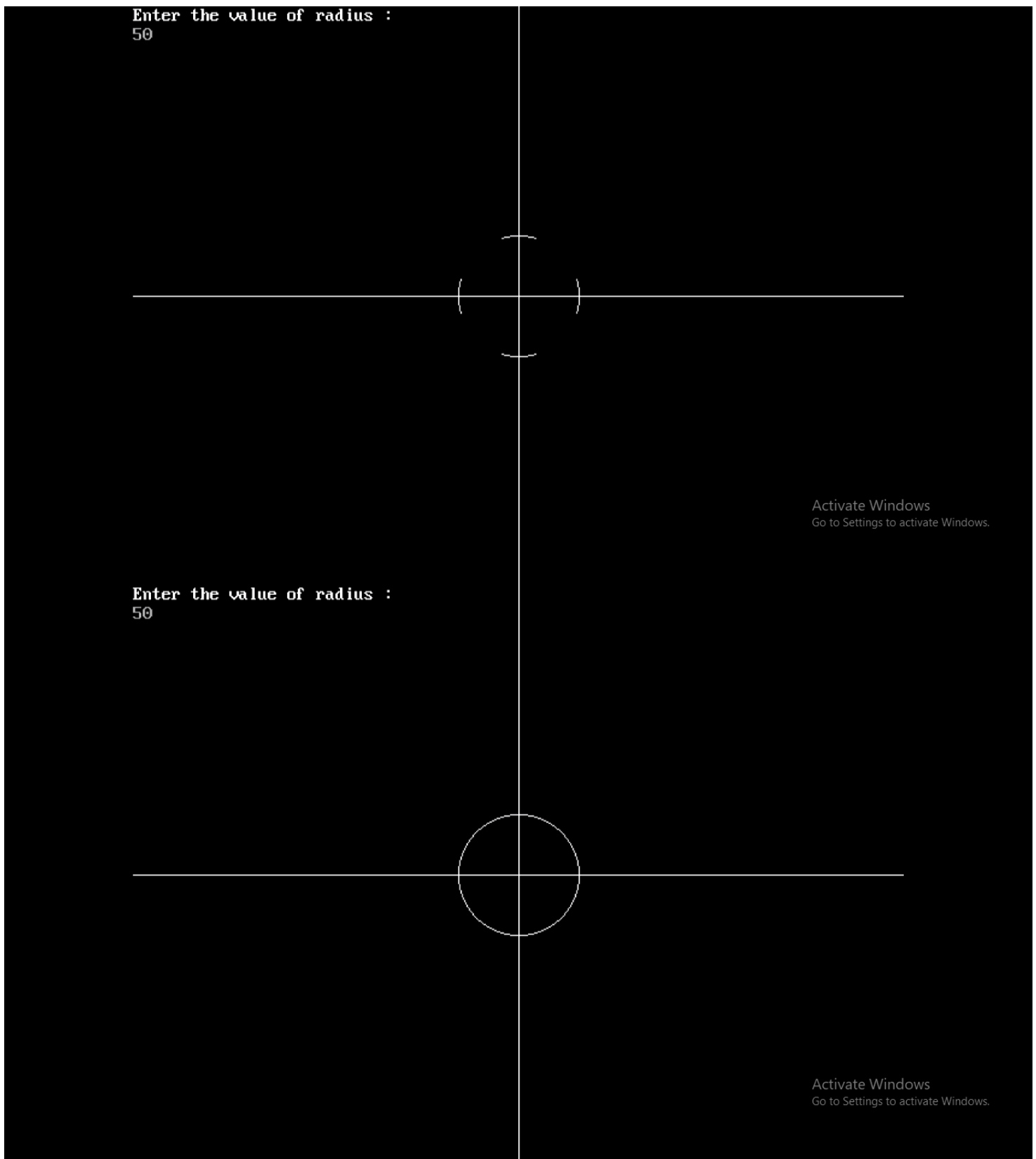
```
File Edit Search Run Compile Debug Project Options Window Help
SYIITMID.C
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
    int xk,xk,x0,y0,k,pk,r,xc,yc;
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "c:\\\\turbooc3\\\\bgi");
    printf("Enter the value of radius :\\n");
    scanf("%d", &r);

    line(320,0,320,640);
    line(0,240,640,240);

    xk = 0;
    yk = r;
    xc = 320;
    yc = 240;

    pk = 1 - r;
    while(xk<yk)
    {
        putpixel(xk + xc, yk + yc, WHITE);
        putpixel(yk + xc, xk + yc, WHITE);
        putpixel(xk + xc, -yk + yc, WHITE);
        putpixel(yk + xc, -xk + yc, WHITE);
        putpixel(-xk + xc, -yk + yc, WHITE);
        putpixel(-yk + xc, -xk + yc, WHITE);
        putpixel(-xk + xc, yk + yc, WHITE);
        putpixel(-yk + xc, xk + yc, WHITE);

        if(pk<0)
        {
            xk = xk + 1;
            yk = yk;
            pk = pk + 2 * (xk) + 1;
        }
        else
        {
            xk = xk + 1;
            yk = yk - 1;
            pk = pk + 2 * (xk) + 1 - 2 * (yk);
        }
        delay(100);
    }
    getch();
    closegraph();
}
```



b. Develop the program for the mid-point ellipse drawing algorithm.