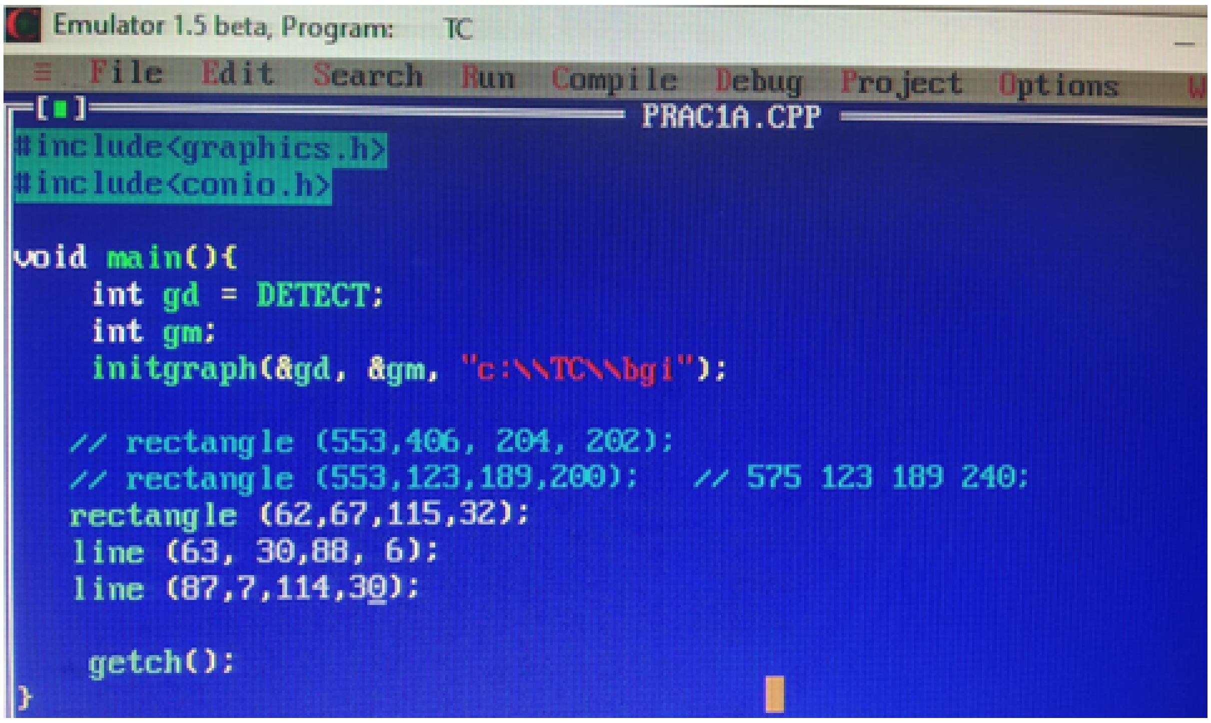
CGA Practical’s

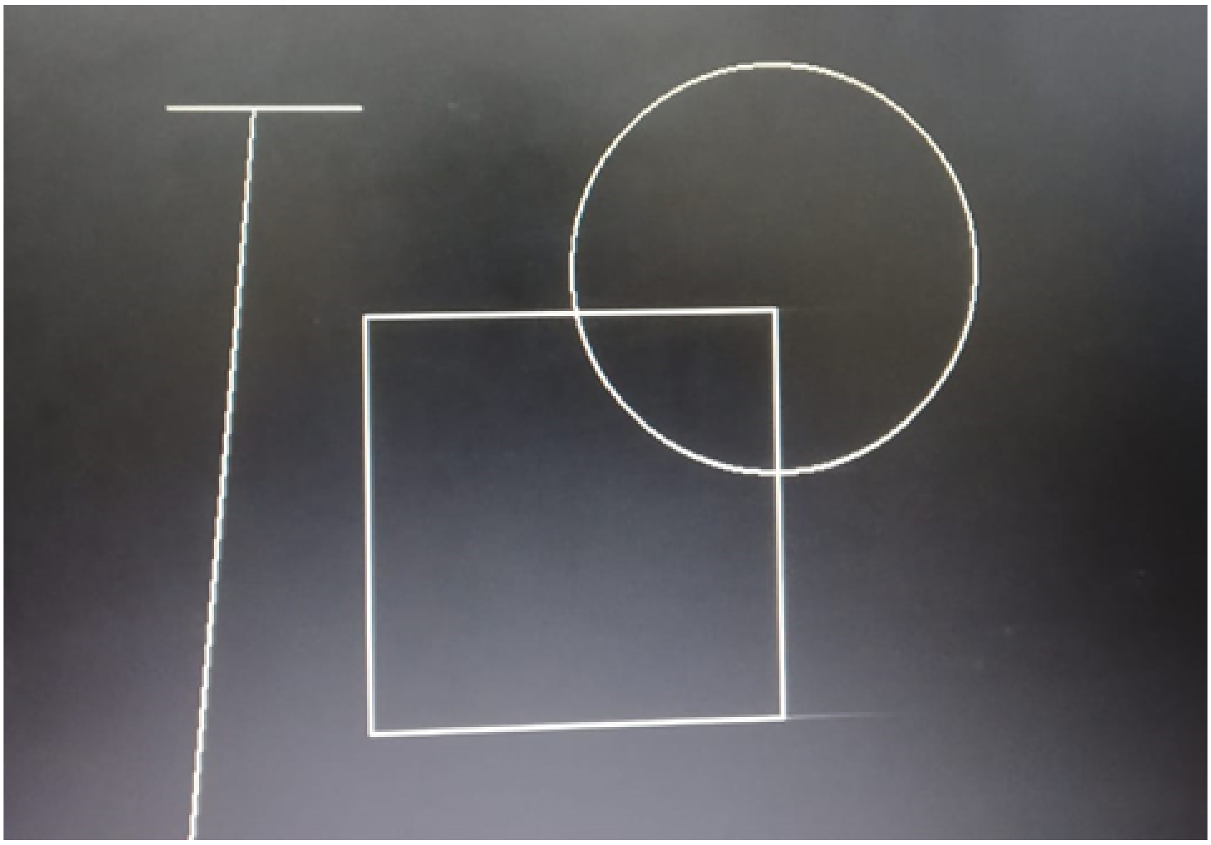
Practical 1 -

1. Drawing a line using pre-defined function +
2. Draw a Square using multiple lines function +
3. Draw Circle, rectangle using pre-defined function -

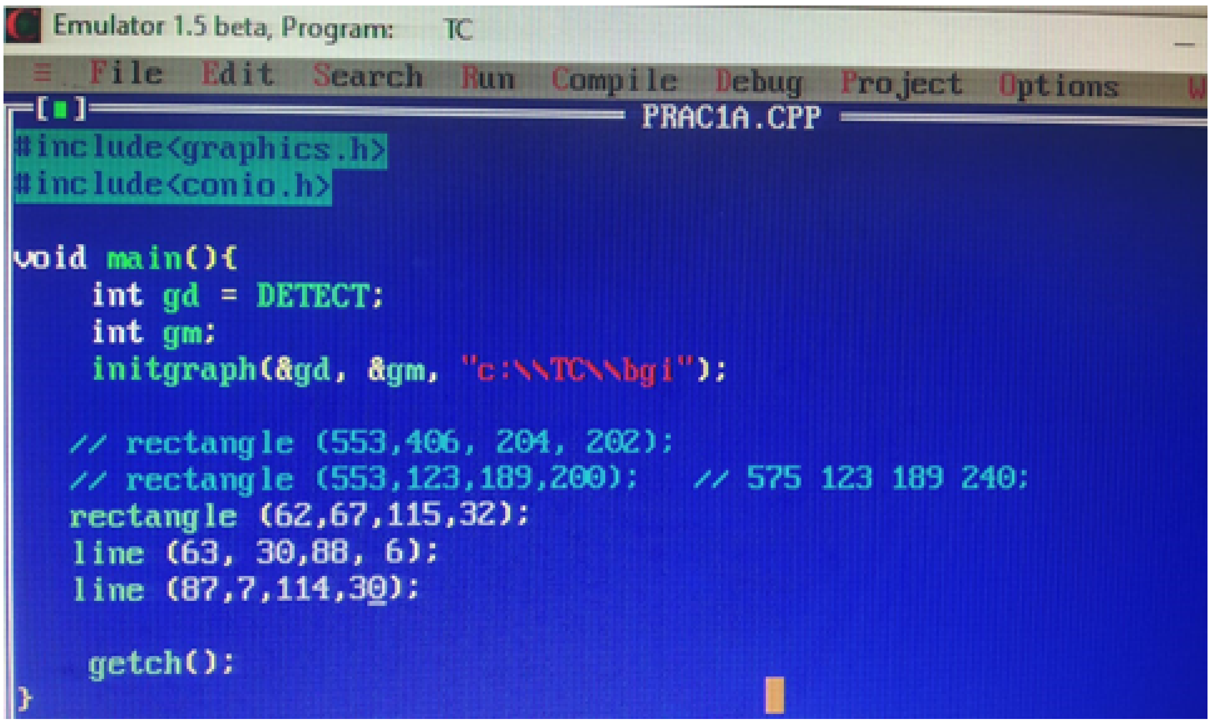


Emulator 1.5 beta, Program: TC
File Edit Search Run Compile Debug Project Options
PRAC1A.CPP

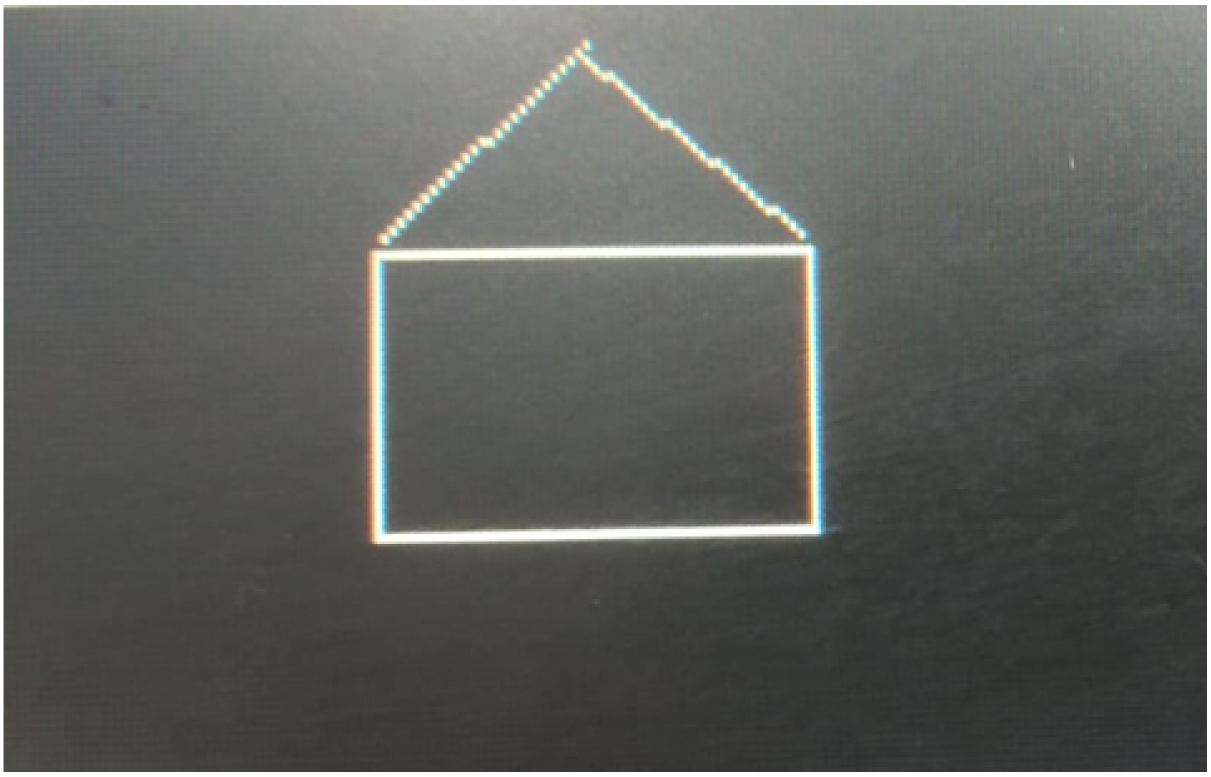
```
#include<graphics.h>
#include<conio.h>

void main(){
     int gd = DETECT;
     int gm;
     initgraph(&gd, &gm, "c:\\TC\\bgi");

    // rectangle (553,406, 204, 202);
    // rectangle (553,123,189,200);   // 575 123 189 240;
    rectangle (62,67,115,32);
    line (63, 30,88, 6);
    line (87,7,114,30);

     getch();
}
```



4. Draw a Hut & add the screen shot

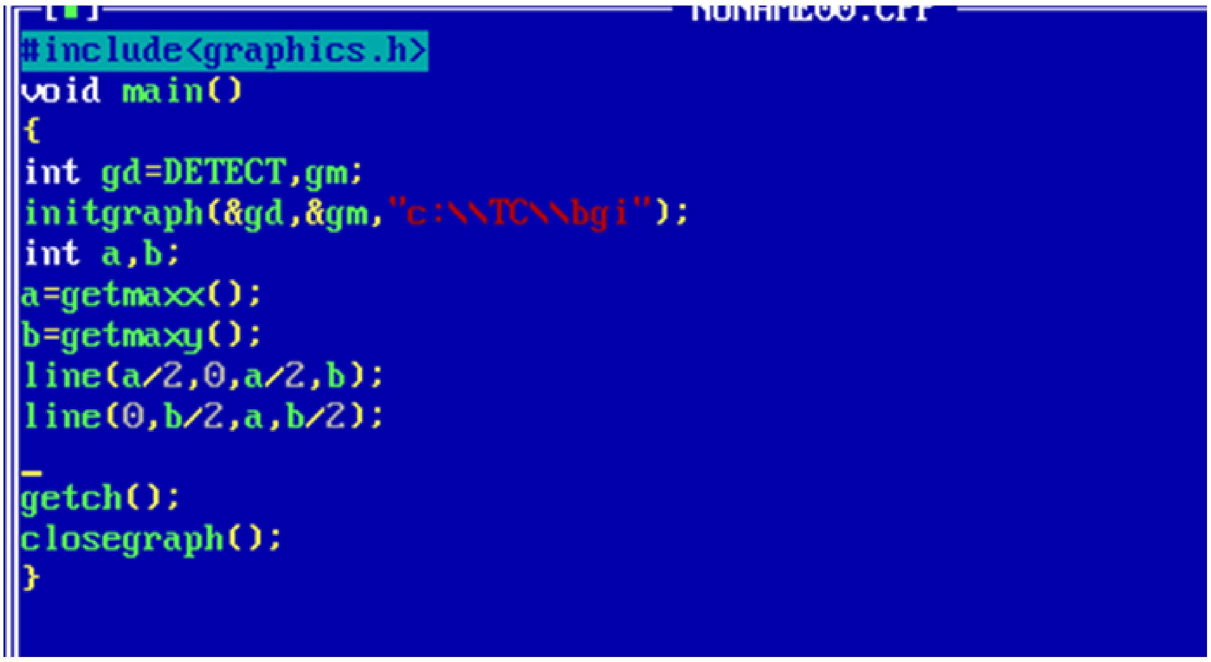

Emulator 1.5 beta, Program: TC
≡ File Edit Search Run Compile Debug Project Options
[■] PRAC1A.CPP

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

void main(){
      int gd = DETECT;
      int gm;
      initgraph(&gd, &gm, "c:\\TC\\bgi");

    // rectangle (553,406, 204, 202);
    // rectangle (553,123,189,200);   // 575 123 189 240;
    rectangle (62,67,115,32);
    line (63, 30,88, 6);
    line (87,7,114,30);

     getch();
}
```

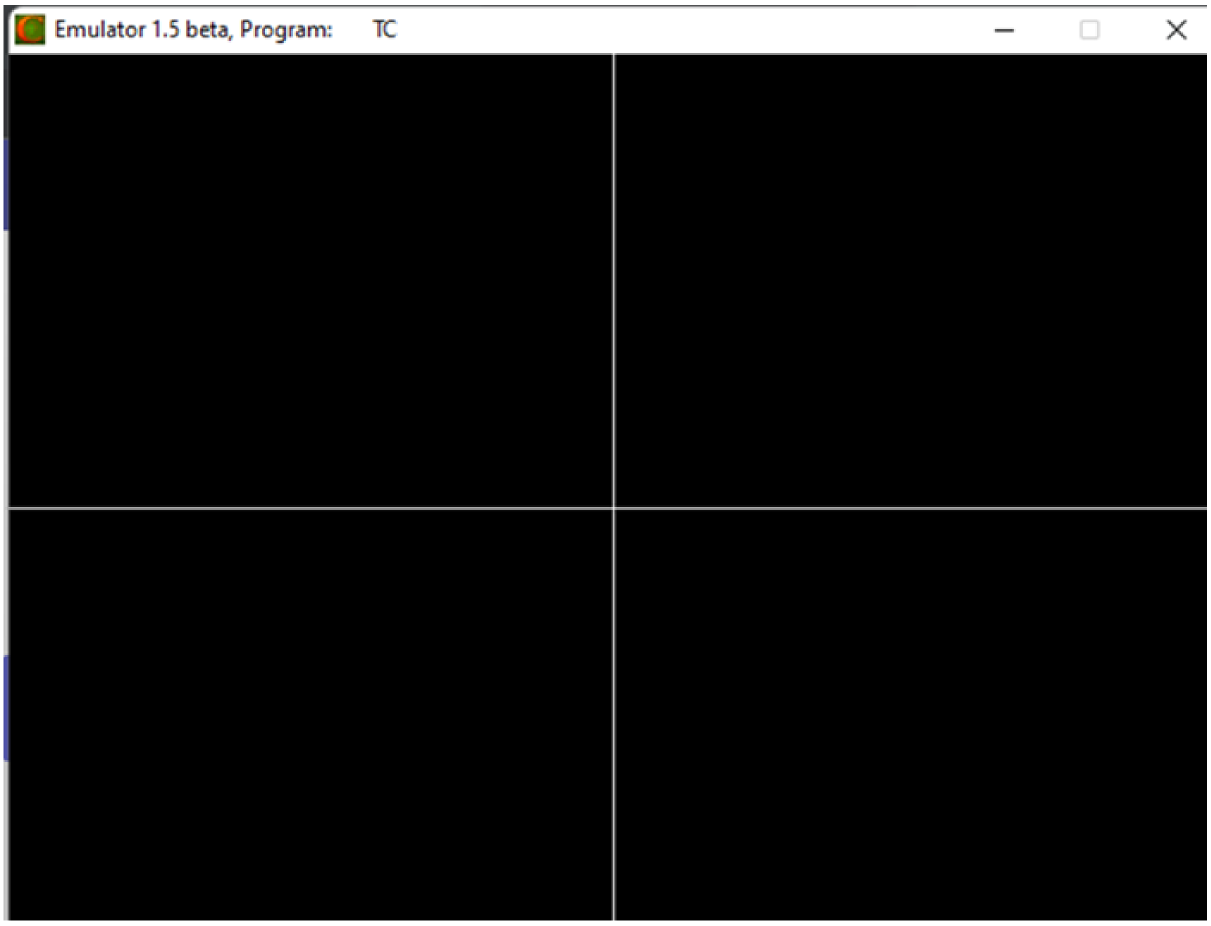


Practical 2 -

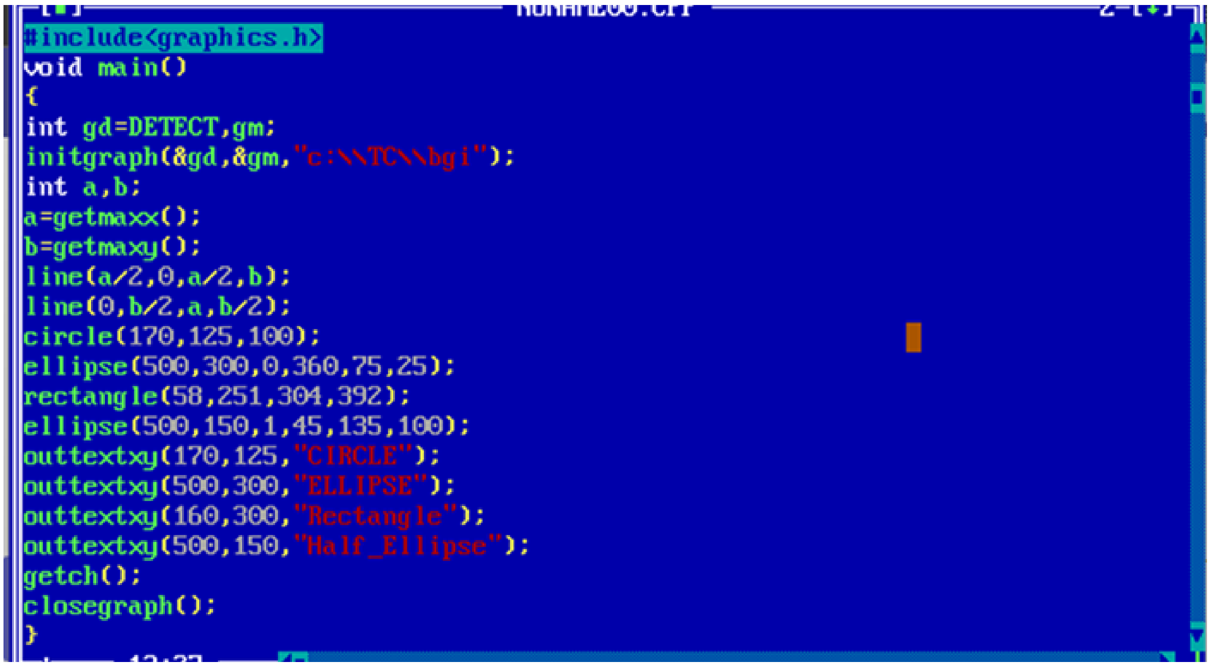
1. Draw a co-ordinate axis at the center of the screen



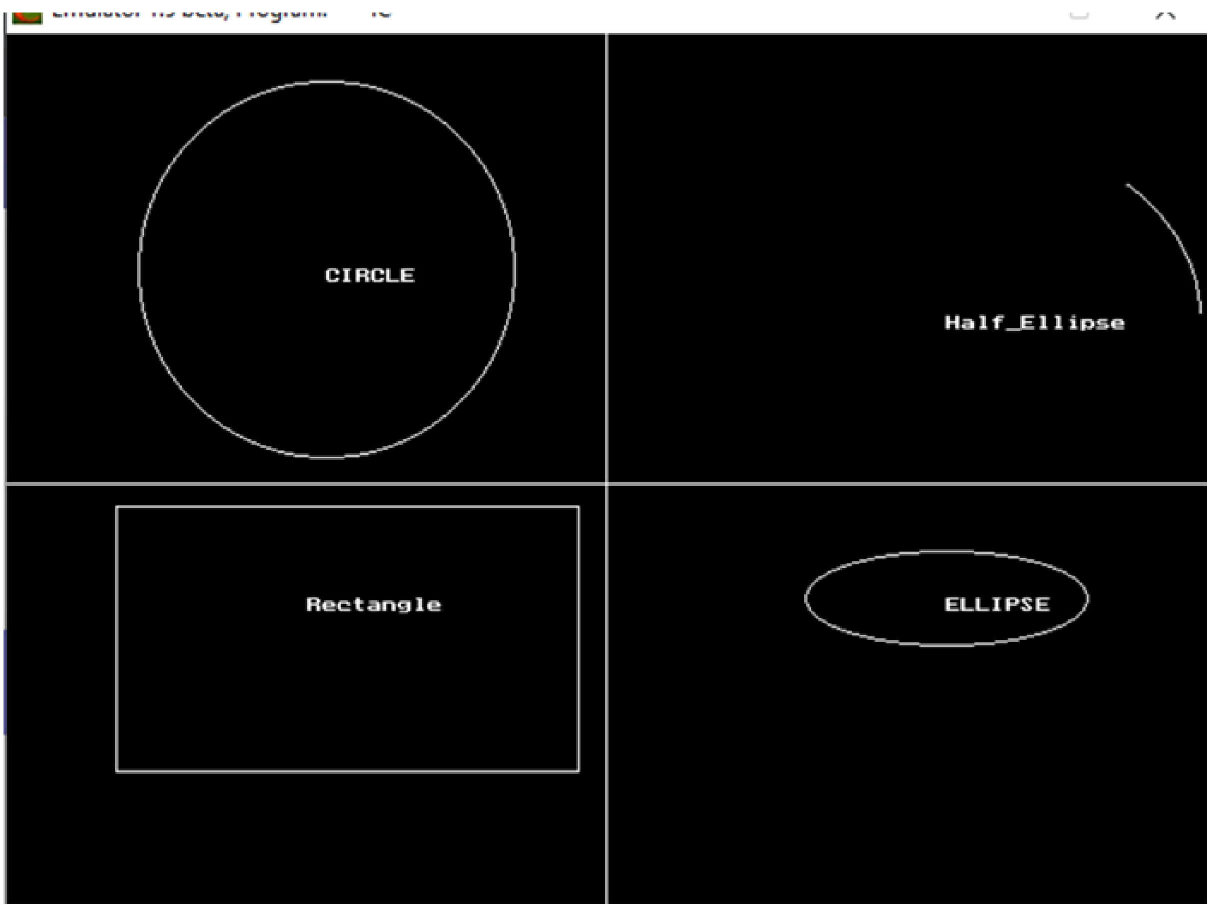
```
#include<graphics.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\TC\\bgi");
int a,b;
a=getmaxx();
b=getmaxy();
line(a/2,0,a/2,b);
line(0,b/2,a,b/2);

getch();
closegraph();
}
```



2. Use above co-ordinate axis and draw Circle, Rectangle, Ellipse, arc in each quadrant.

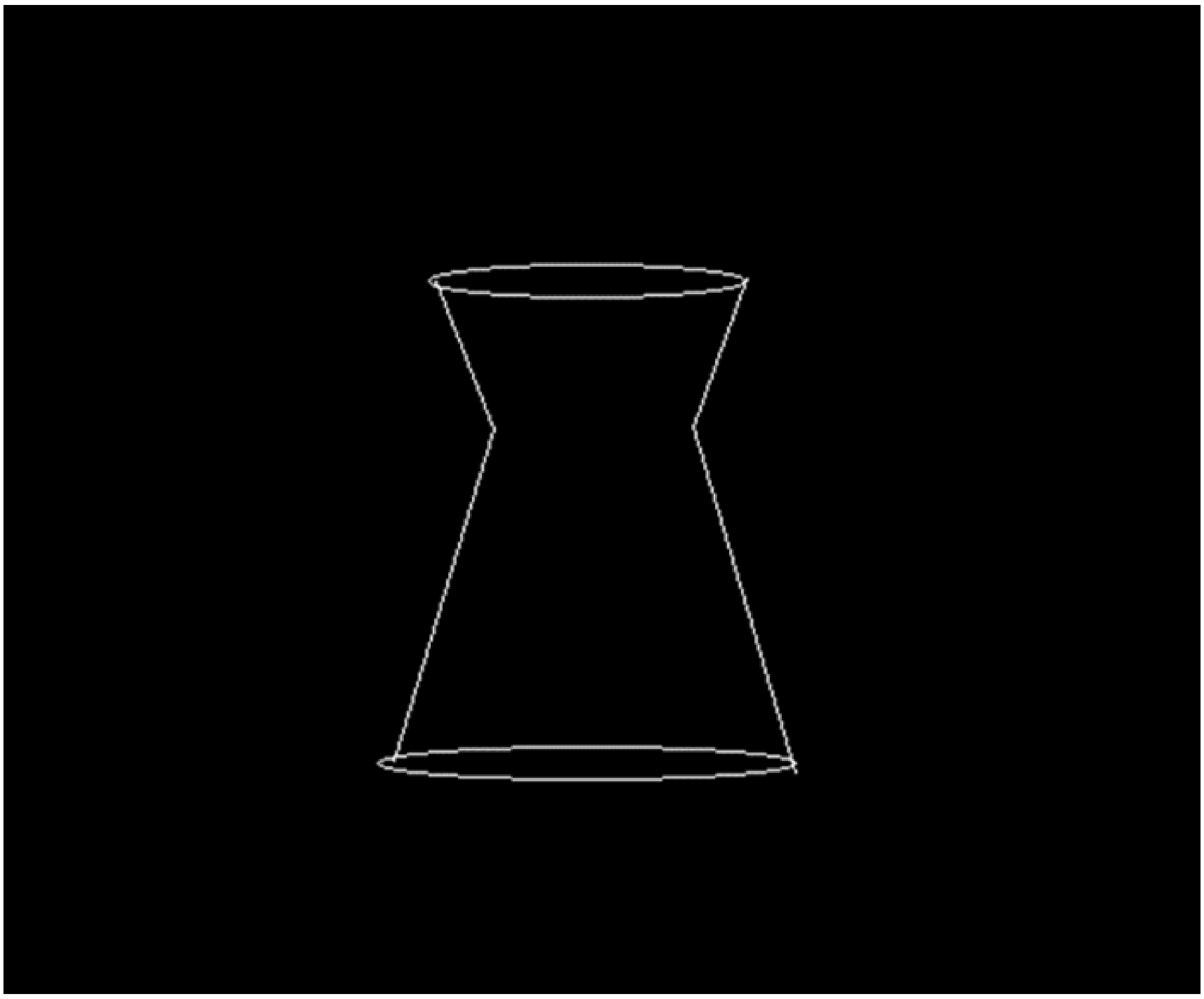


```
#include<graphics.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\TC\\bgi");
int a,b;
a=getmaxx();
b=getmaxy();
line(a/2,0,a/2,b);
line(0,b/2,a,b/2);
circle(170,125,100);
ellipse(500,300,0,360,75,25);
rectangle(58,251,304,392);
ellipse(500,150,1,45,135,100);
outtextxy(170,125,"CIRCLE");
outtextxy(500,300,"ELLIPSE");
outtextxy(160,300,"Rectangle");
outtextxy(500,150,"Half_Ellipse");
getch();
closegraph();
}
```



3. Draw flower-pot using different shapes

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

int main(){
    int gd = DETECT;
    int gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");

    line(310,172,335,236);
    line(444,171,421,235);
    line(335,236,292,379);
    line(421,235,465,384);
    ellipse(375,380,0,360,90,7);
    ellipse(375,172,0,360,68,7);

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

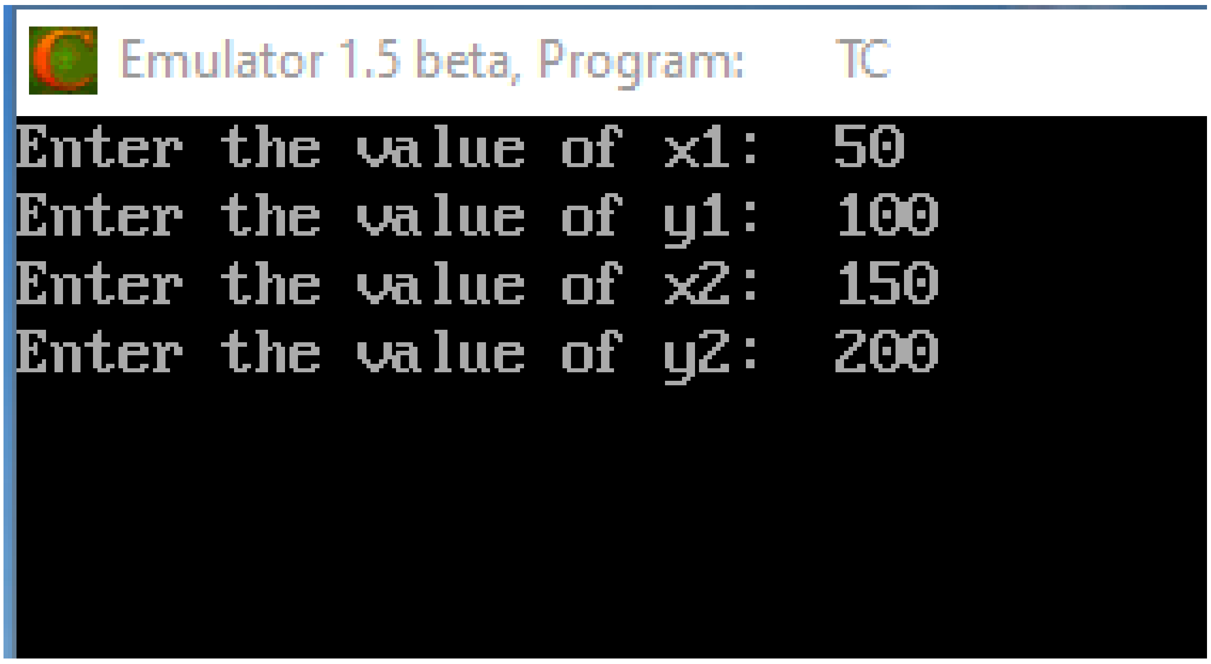


Practical 3 -

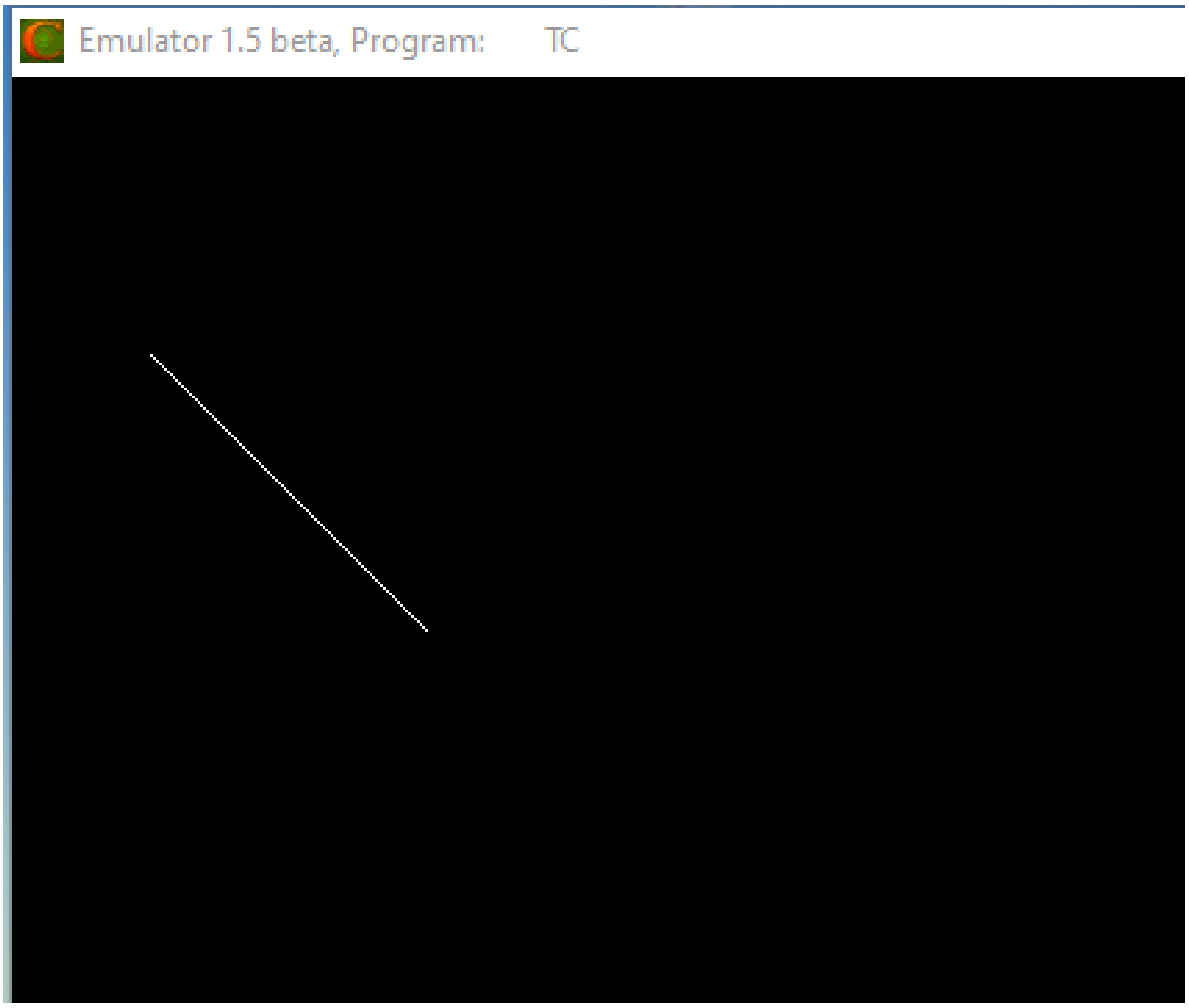
2. Consider the 2 end points of a line as (50,50), (100,100). Draw the line using DDA Line drawing algorithm

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

void main(){

    float x,y,x1,y1,x2,y2,dx,dy,length;
     int i,gd=DETECT,gm;
     initgraph(&gd,&gm,"C:\\TC\\BGI");

    cout<<"Enter value of x1: \t";
     cin>>x1;
     cout<<"Enter value of y1: \t";
```

```
      cin>>y1;

    cout<<"Enter value of x2: \t";
      cin>>x2;
      cout<<"Enter value of y2: \t";
      cin>>y2;

      dx=abs(x2-x1);
      dy=abs(y2-y1);

    if(dx>=dy)
      {
        length=dx;
      }
      else
      {
        length=dy;
      }

      dx=(x2-x1)/length;
       dy=(y2-y1)/length;

     x=x1+0.5;
       y=y1+0.5;
       i=1;

     while(i<=length)
       {
         putpixel(x,y,15);
         x=x+dx;
         y=y+dy;
         i=i+1;
         delay(100);
       }

     getch();
       closegraph();

 }
```

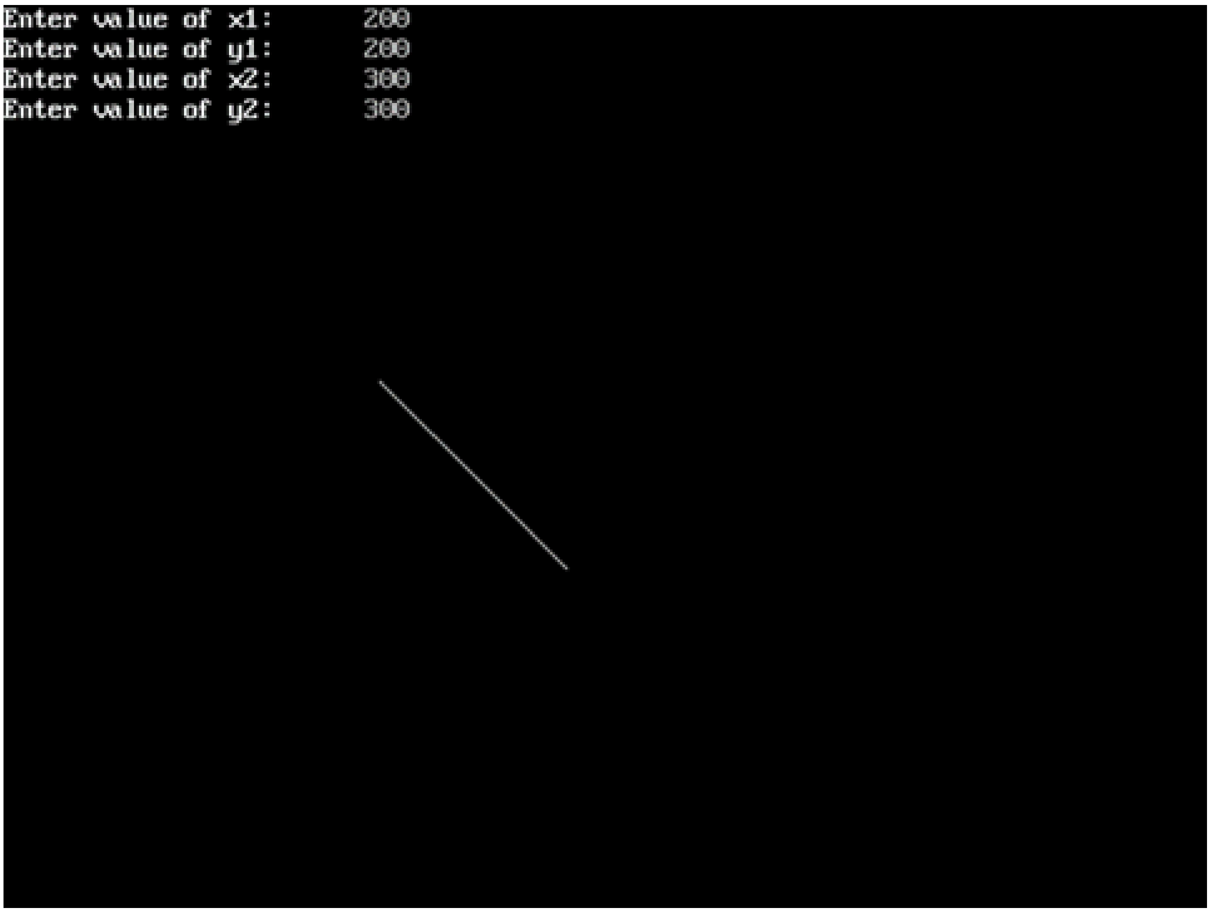


```
Enter the value of x1:  50
Enter the value of y1:  100
Enter the value of x2:  150
Enter the value of y2:  200
```



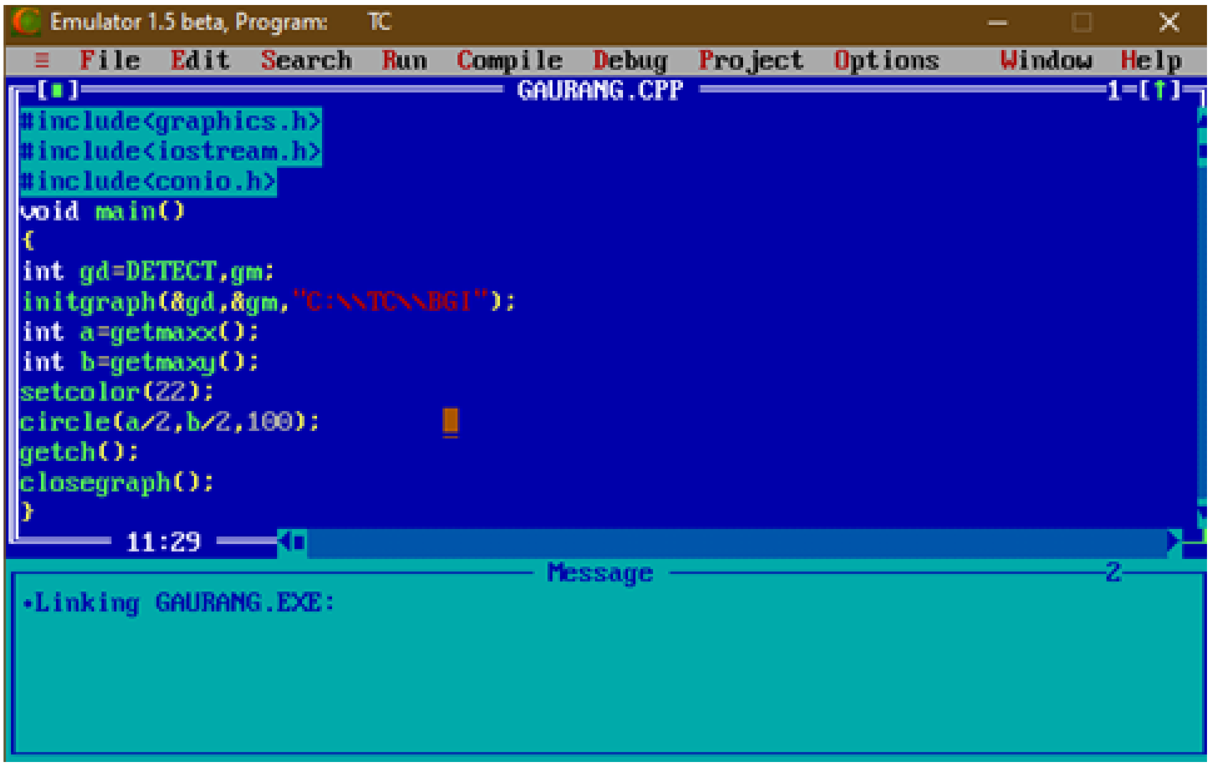
2. Consider the 2 end points of a line as (70,50), (120,150). Draw the line using Bresenham’s Line drawing algorithm.

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

void main(){

   float x,y,x1,y1,dx,dy,e,x2,y2;
     int i,gd=DETECT,gm;
     initgraph(&gd,&gm,"c:\\tc\\bgi");

   cout<<"Enter value of x1: \t";
     cin>>x1;
     cout<<"Enter value of y1: \t";
     cin>>y1;

   cout<<"Enter value of x2: \t";
     cin>>x2;
     cout<<"Enter value of y2: \t";
     cin>>y2;


   dx=(x2-x1);
     dy=(y2-y1);

   x=x1;
     y=y1;
     e=2*dy-dx;
     i=1;

   do{
     putpixel(x,y,15);


   while(e>=0)
     {
        y=y+1;
        e=e-2*dx;
     }
     x=x+1;
     e=e+2*dy;
     i=i+1;
     }while(i<=dx);

     getch();
      closegraph();

}
```



```
Enter value of x1:      200
Enter value of y1:      200
Enter value of x2:      300
Enter value of y2:      300
```

3. Draw basic shapes in the centre of the screen. (Circle, line, rectangle, ellipse, arc, concentric circles, square, pentagon, Hexagon, Star).


Emulator 1.5 beta, Program: TC
≡ File Edit Search Run Compile Debug Project Options Window Help
GAURANG.CPP
11:29
Message

```
#include<graphics.h>
#include<iostream.h>
#include<conio.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"C:\\TC\\BGI");
int a=getmaxx();
int b=getmaxy();
setcolor(22);
circle(a/2,b/2,100);
getch();
closegraph();
}
```

```
•Linking GAURANG.EXE:
```

Practical 4 -

1. Consider the 2 end points of a line as (70,50), (120,150). Draw the line using Bresenham's Line drawing algorithm.

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

void main(){
int gd=DETECT,gm;
initgraph(&gd, &gm, "C:\\TC\\BGI");
int x0=70, y0 = 50 , x1 = 120, y1 = 150 , dx,dy,p,x,y;
dx=x1-x0;
dy=y1-y0;
y=y0;
p = 2 *dy-dx;

while(x<x1){
 if (p>=0)
 {

putpixel(x,y,7);

y=y+1;
y=y0;

p=2*dy-dx;
while(x<x1){
  if (p>=0){
  putpixel(x,y,7);
  y=y+1;
  p=p+2*dy-2*dx;
  }
else{
 putpixel(x,y,7);
  p=p+2*dy; }
 x=x+1;
 }

getch();

}
}
}
```

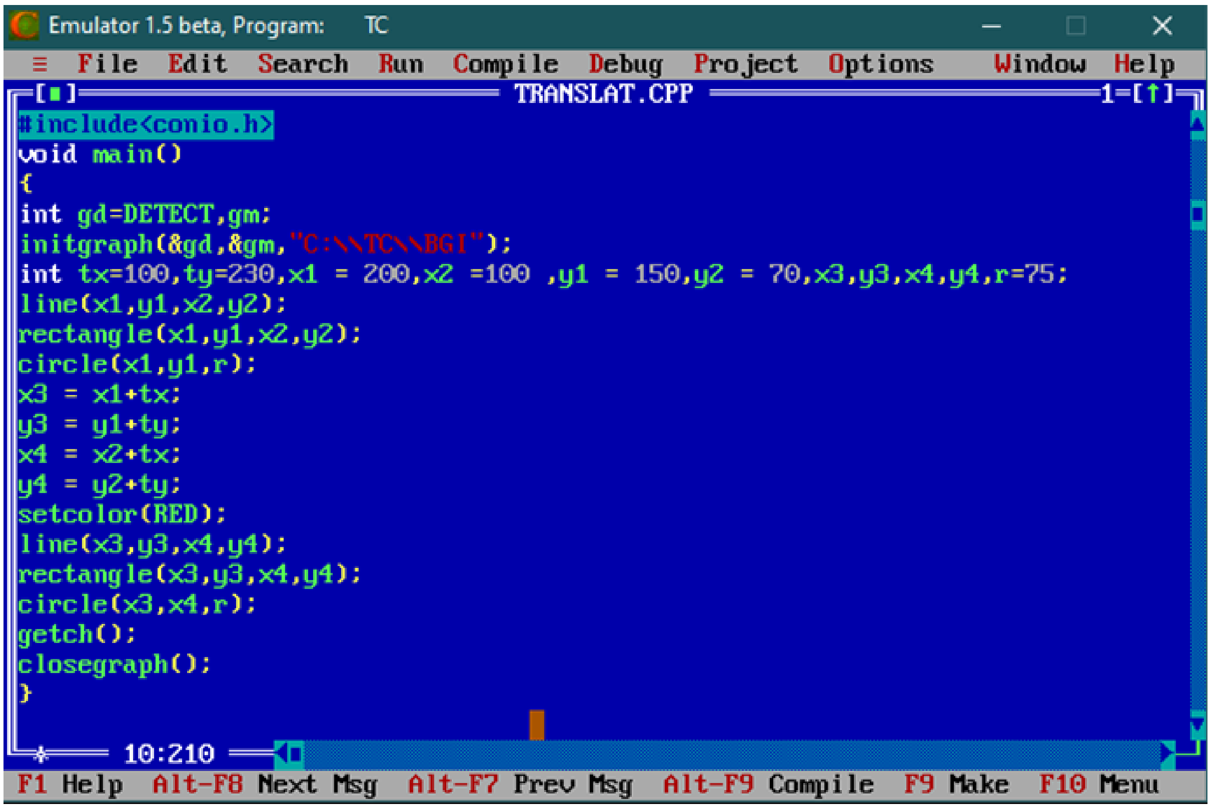
2. Implement Mid-point circle drawing algorithm with radius 70.

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

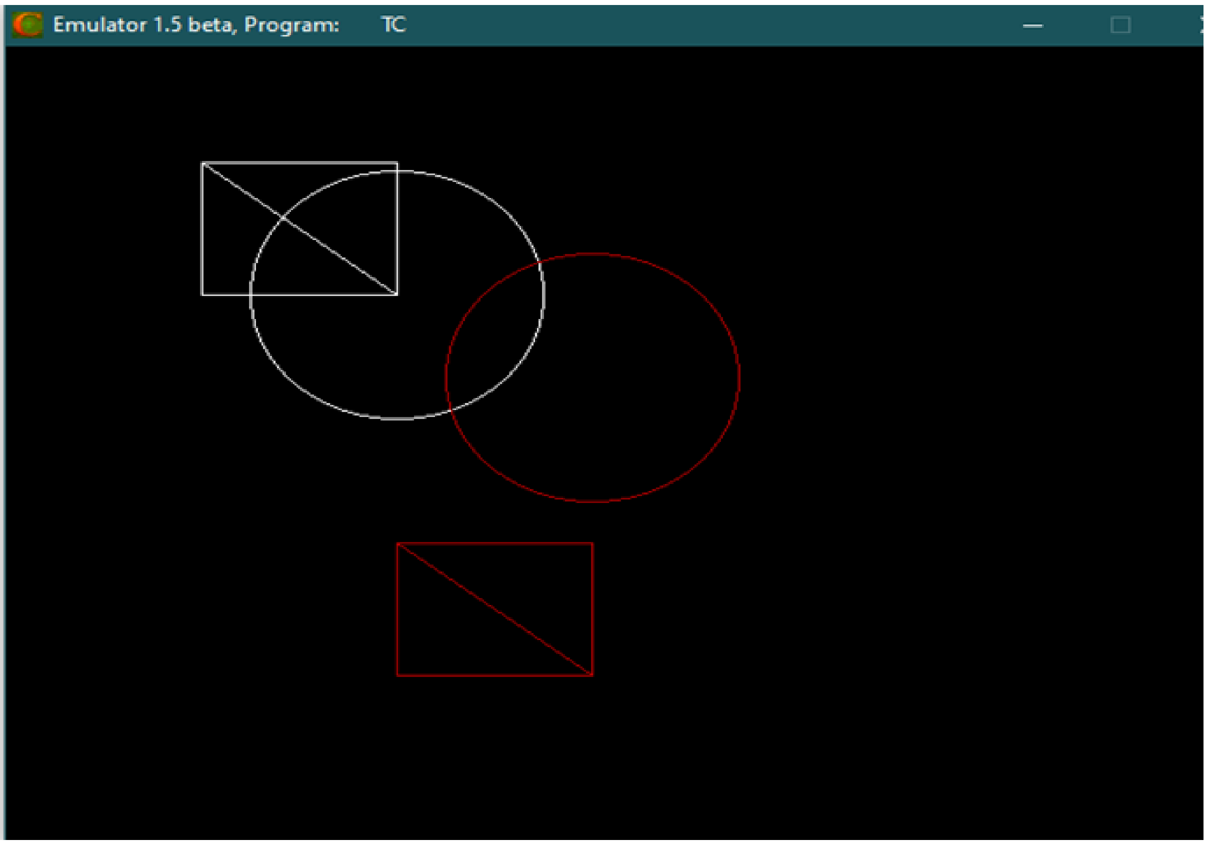
```
#include<graphics.h>
#include<conio.h>
#include<math.h>

void main(){
float d;
int gd=DETECT,gm,x,y,r;
initgraph(&gd,&gm,"C:\\TC\\BGI");

cout<<"Enter the radius of a circle :";
cin>>r;

x = 0;
y = r;

d = (5/4) - r;
do{
  putpixel(200+x,200+y,15);
  putpixel(200+y,200+x,13);
  putpixel(200+y,200-x,11);
  putpixel(200+x,200-y,9);
  putpixel(200-x,200-y,7);
  putpixel(200-y,200-x,5);
  putpixel(200-y,200+x,3);
  putpixel(200-x,200+y,1);

if (d < 0){
  d = d + 2*x + 3;
}
else{
  d = d + 2*(x-y) + 5;
  y = y - 1;
}
x = x + 1;
delay(10);
} while(x < y);


getch();
closegraph();
}
```

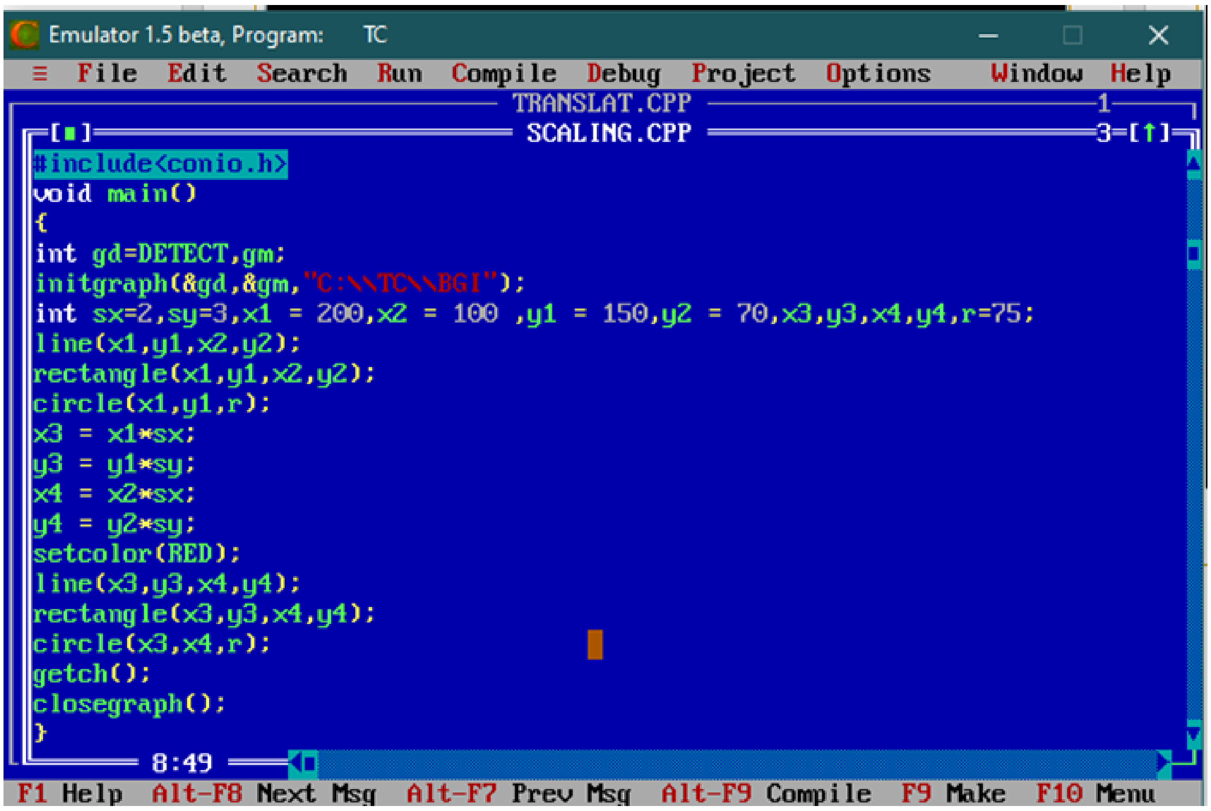
Practical 5 -

1. Demonstrate 2D transformation - Translation, Scaling, Rotation (for different objects like point, line rectangle, circle)

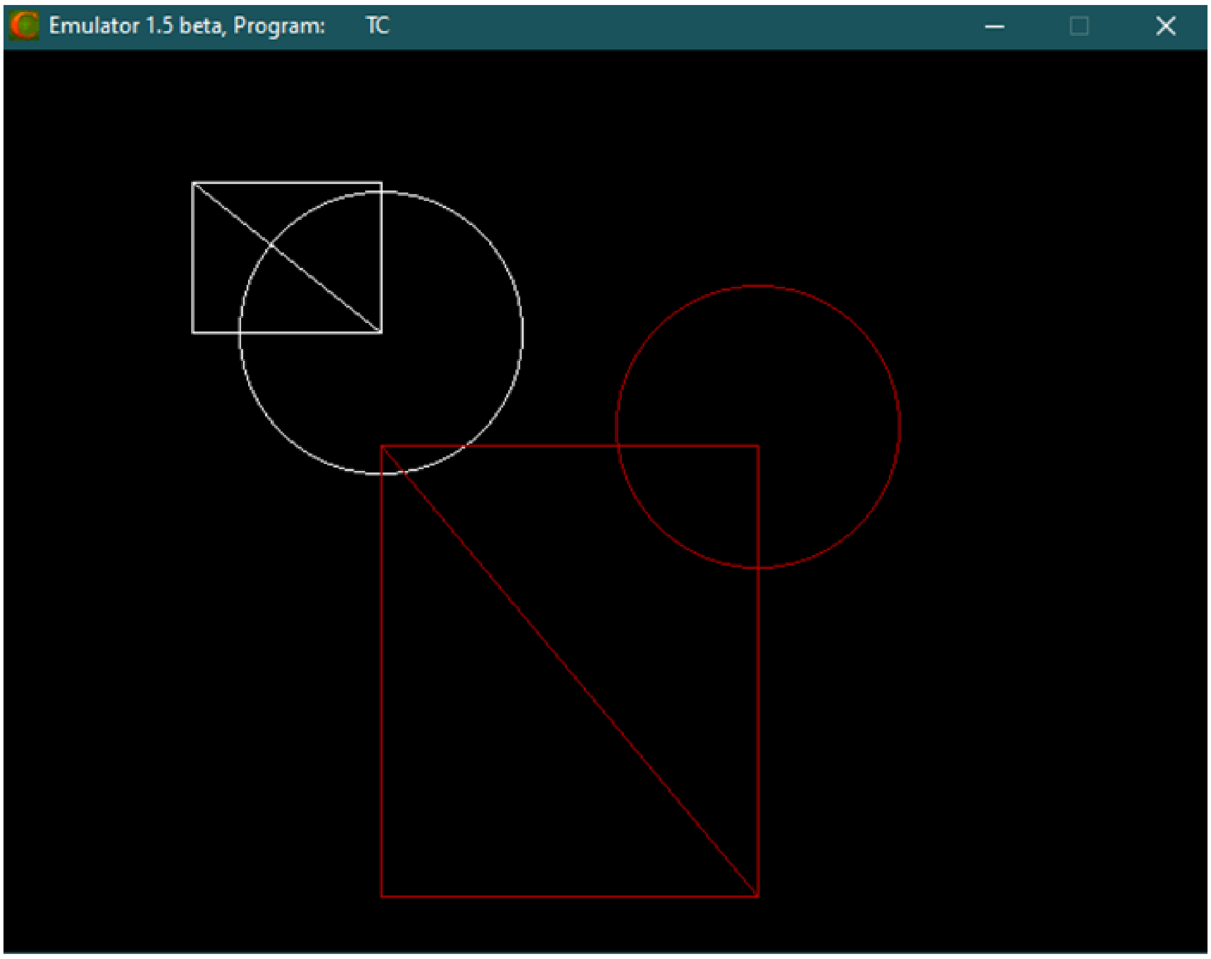

Emulator 1.5 beta, Program: TC
≡ File Edit Search Run Compile Debug Project Options Window Help
TRANSLAT.CPP
10:210
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

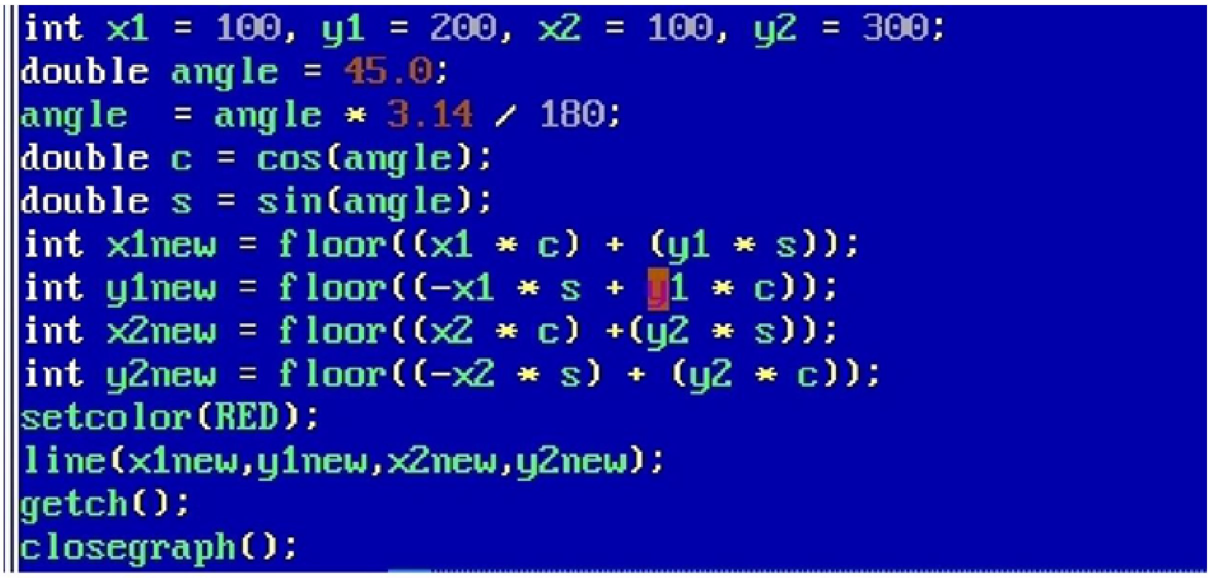
```
#include<conio.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"C:\\TC\\BGI");
int tx=100,ty=230,x1 = 200,x2 =100 ,y1 = 150,y2 = 70,x3,y3,x4,y4,r=75;
line(x1,y1,x2,y2);
rectangle(x1,y1,x2,y2);
circle(x1,y1,r);
x3 = x1+tx;
y3 = y1+ty;
x4 = x2+tx;
y4 = y2+ty;
setcolor(RED);
line(x3,y3,x4,y4);
rectangle(x3,y3,x4,y4);
circle(x3,x4,r);
getch();
closegraph();
}
```



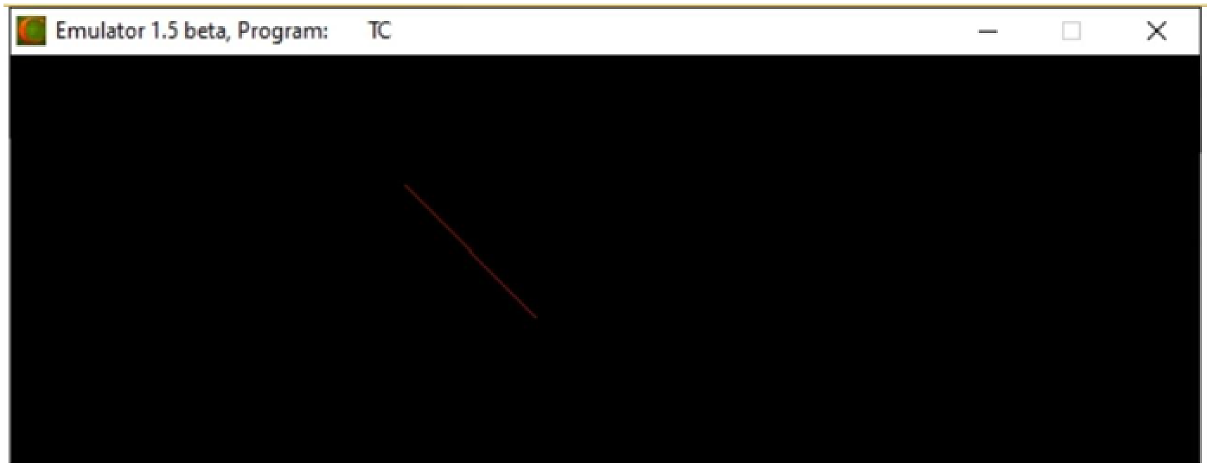

Emulator 1.5 beta, Program: TC
≡ File Edit Search Run Compile Debug Project Options Window Help
TRANSLAT.CPP
SCALING.CPP
8:49
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
#include<conio.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"C:\\TC\\BGI");
int sx=2,sy=3,x1 = 200,x2 = 100 ,y1 = 150,y2 = 70,x3,y3,x4,y4,r=75;
line(x1,y1,x2,y2);
rectangle(x1,y1,x2,y2);
circle(x1,y1,r);
x3 = x1*sx;
y3 = y1*sy;
x4 = x2*sx;
y4 = y2*sy;
setcolor(RED);
line(x3,y3,x4,y4);
rectangle(x3,y3,x4,y4);
circle(x3,x4,r);
getch();
closegraph();
}
```





```
int x1 = 100, y1 = 200, x2 = 100, y2 = 300;
double angle = 45.0;
angle  = angle * 3.14 / 180;
double c = cos(angle);
double s = sin(angle);
int x1new = floor((x1 * c) + (y1 * s));
int y1new = floor((-x1 * s + y1 * c));
int x2new = floor((x2 * c) +(y2 * s));
int y2new = floor((-x2 * s) + (y2 * c));
setcolor(RED);
line(x1new,y1new,x2new,y2new);
getch();
closegraph();
```


Emulator 1.5 beta, Program: TC