Assignment 6 a :-

Code :-

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

using namespace std;

class algo
{
   float r,x2,y2,x3,y3,dx,dy,len,len1,len2,dx1,dy1,dx2,dy2;
   int i;
   float x,y,ax,ay,cx,cy,a1,a2,a3,b1,b2,b3,c1,d1;
   public:
      void triangle(float x1, float y1)
      {

         float sx,qx,qy,sy;
         a1=sx=x1;
         b1=sy=y1;
         a2=x2=x1-150;
         b2=y2=y1+260;
         a3=x3=sx+150;
```

```
b3=y3=sy+260;
dx=x2-x1;
dy=y2-y1;
dx1=x3-x2;
dy1=y3-y2;
dx2=x3-sx;
dy2=y3-sy;

   if(abs(dx)>=abs(dy))
      len=abs(dx);
   else
      len=abs(dy);
   x=dx/len;
   y=dy/len;

   for(i=1;i<len;i++)
   {
      putpixel(x1,y1,15);
      x1=x1+x;
      y1=y1+y;
   }
```

```
//----------------line 2
if(abs(dx1)>=abs(dy1))
   len1=abs(dx1);
else
   len1=abs(dy1);
ax=dx1/len1;
ay=dy1/len1;

for(i=1;i<len1;i++)
{
   putpixel(x2,y2,15);
   x2=x2+ax;
   y2=y2+ay;
}

//---------------line 3
if(dx2>=dy2)
   len2=abs(dx2);
else
   len2=abs(dy2);
cx=dx2/len2;
cy=dy2/len2;
```

```
                for(i=1;i<len2;i++)
                {
                    putpixel(sx,sy,WHITE);
                    sx=sx+cx;
                    sy=sy+cy;
                }
        }

        void incircle()
        {
            float mx,my,d;
            c1=(a1+a2+a3)/3;
            d1=(b1+b2+b3)/3;
//cout<<c1<<d1;
            c1--;
            r=85;
            mx=0;
            my=r;
            d=3-(2*r);

            while(mx<my)
```

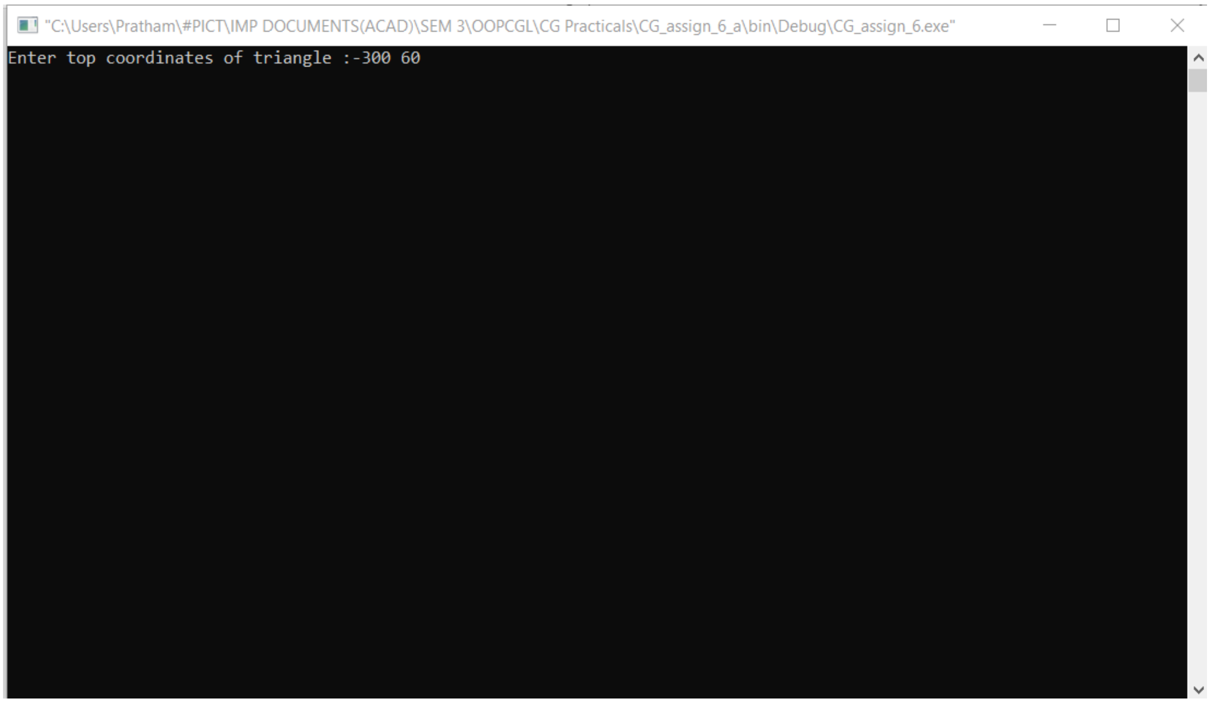
```
{
   putpixel(mx+c1,my+d1,15);
   delay(1);
   putpixel(mx+c1,d1-my,15);
   delay(1);
   putpixel(c1-mx,my+d1,15);
   delay(1);
   putpixel(c1-mx,d1-my,15);
   delay(1);
   putpixel(c1+my,d1+mx,15);
   delay(1);
   putpixel(my+c1,d1-mx,15);
   delay(1);
   putpixel(c1-my,d1+mx,15);
   delay(1);
   putpixel(c1-my,d1-mx,15);
   delay(1);
   if(d<=0)
   {
      d=d+(4*mx)+6;
      mx++;
   }
```

```
        else
        {
            d=d+(4*(mx-my))+10;
            mx++;
            my--;
        }
    }
}

void circumcircle()
{
    float mx,my,d,r1;
    c1=(a1+a2+a3)/3;
    d1=(b1+b2+b3)/3;

    r1=173.33;
    mx=0;
    my=r1;
    d=3-(2*r1);

    while(mx<my)
    {
```

```
putpixel(mx+c1,my+d1,15);
delay(1);
putpixel(mx+c1,d1-my,15);
delay(1);
putpixel(c1-mx,my+d1,15);
delay(1);
putpixel(c1-mx,d1-my,15);
delay(1);
putpixel(c1+my,d1+mx,15);
delay(1);
putpixel(my+c1,d1-mx,15);
delay(1);
putpixel(c1-my,d1+mx,15);
delay(1);
putpixel(c1-my,d1-mx,15);
delay(1);
if(d<=0)
{
   d=d+(4*mx)+6;
   mx++;
}
else
```

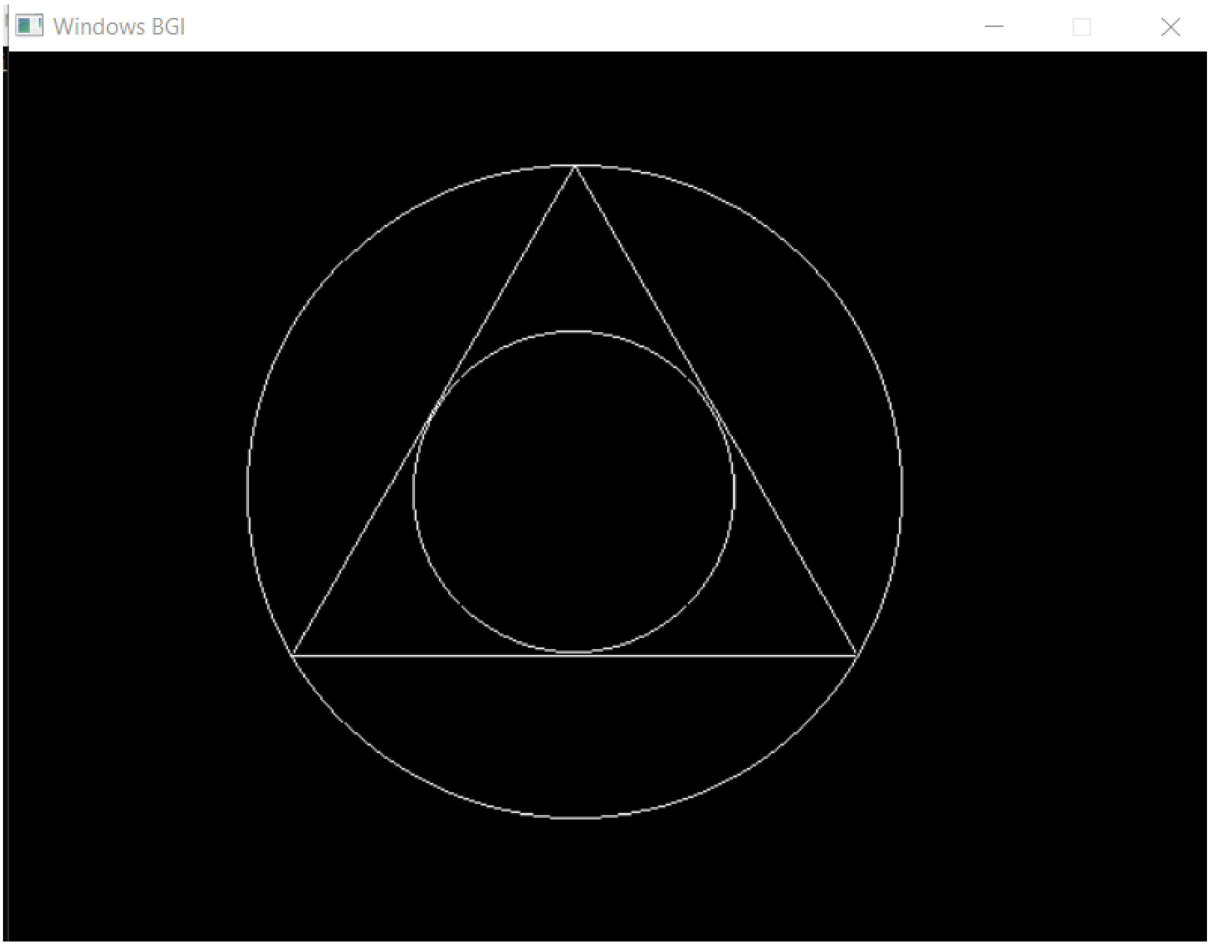
```
                {
                    d=d+(4*(mx-my))+10;
                    mx++;
                    my--;
                }
            }
        }
}s;

int main()
{



    int gd,gm;
    float x1,y1;
    gd=DETECT;

    cout<<"Enter top coordinates of triangle :-";

    cin>>x1>>y1;

    initgraph(&gd,&gm,NULL);
    s.triangle(x1,y1);     //Calling the triangle drwaing function
    s.incircle();
    s.circumcircle();


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

Output :-



"C:\Users\Pratham\#PICT\IMP DOCUMENTS(ACAD)\SEM 3\OOPCGL\CG Practicals\CG_assign_6_a\bin\Debug\CG_assign_6.exe"

```
Enter top coordinates of triangle :-300 60
```


Windows BGI

Assignment 6 b :-

Code :-

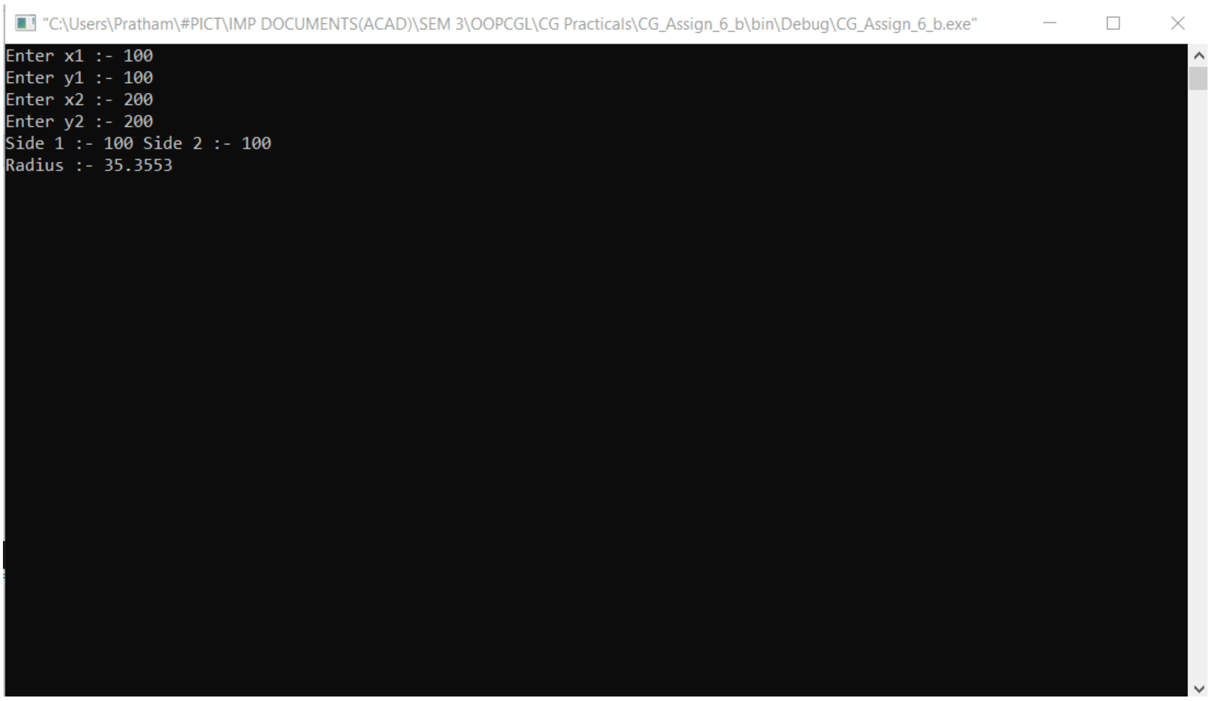
```
#include<iostream>

#include<graphics.h>

#include<math.h>


using namespace std;


class drawpattern

{

private:

float dx,dy,i ,length;

float count;

public:

int x1,y1,x2,y2;

int xmid,ymid;

void getdata();

void ddaline(int x1,int x2,int y1, int y2);

 int xc,yc,r;

 void bdrawcircle(int xc,int yc,int r);

};

void drawpattern::getdata()

{

cout<<"Enter x1 :- ";
```

```
cin>>x1;

cout<<"Enter y1 :- ";

cin>>y1;

cout<<"Enter x2 :- ";

cin>>x2;

cout<<"Enter y2 :- ";

cin>>y2;

}

void drawpattern::ddaline(int x1, int x2, int y1, int y2)

{

float x,y;

dx = (x2-x1);

dy = (y2-y1);


if(abs(dx)>=abs(dy)) length = abs(dx);

else length = abs(dy);


dx = dx/length;

dy = dy/length;

x=x1;

y=y1;

i=1;


while(i<=length){
```

```
x = x + dx;
y = y + dy;

putpixel(x,y,15);
i++;
}
}
void drawpattern::bdrawcircle(int xc,int yc,int r)
{
//xc=320;
//yc=240;
 int x,y,d;
x=0;
y=r;
putpixel(xc+x,yc-y,15);
// initialize the decision variable
d=3-2*r;
do
{
putpixel(xc+x,yc+y,15);
putpixel(xc-x,yc-y,15);
putpixel(xc+x,yc-y,15);
putpixel(xc-x,yc+y,15);
putpixel(xc+y,yc-x,15);
putpixel(xc-y,yc-x,15);
putpixel(xc+y,yc+x,15);
putpixel(xc-y,yc+x,15);
if(d<0)
{
y=y;
d=d+4*x+6;
}
else
{
d=d+4*(x-y)+10;
y=y-1;
}
x=x+1;
}
while(x<=y);
}
int main()
{

int gdriver= DETECT, gmode;
initgraph(&gdriver,&gmode,"c://Turboc3//BGI");
cleardevice();
 drawpattern d;
 d.getdata();
d.ddaline(d.x1,d.y1,d.x2,d.y1);// (x1,y1) and (x2,y1)
 d.ddaline(d.x2,d.y1,d.x2,d.y2);
 d.ddaline(d.x2,d.y2,d.x1,d.y2);
 d.ddaline(d.x1,d.y2,d.x1,d.y1);
 d.xmid=abs((d.x1+d.x2))/2;
 d.ymid=abs((d.y1+d.y2))/2;
d.ddaline(d.xmid,d.y1,d.x2,d.ymid);// (x1,y1) and (x2,y1)
 d.ddaline(d.x2,d.ymid,d.xmid,d.y2);
 d.ddaline(d.xmid,d.y2,d.x1,d.ymid);
 d.ddaline(d.x1,d.ymid,d.xmid,d.y1);
 float rad,cal,sidex,sidey;
 sidex=abs(d.x2-d.x1);
 sidey=abs(d.y2-d.y1);
 cal=pow(sidex,2)+pow(sidey,2);
 cal=2*sqrt(cal);
 rad=(sidex*sidey)/cal;
 cout<<"Side 1 :- "<<sidex<<" "<<"Side 2 :- "<<sidey;
 cout<<"\nRadius :- "<<rad;
d.bdrawcircle(d.xmid,d.ymid,rad);
getch();
closegraph();

return 0;
```

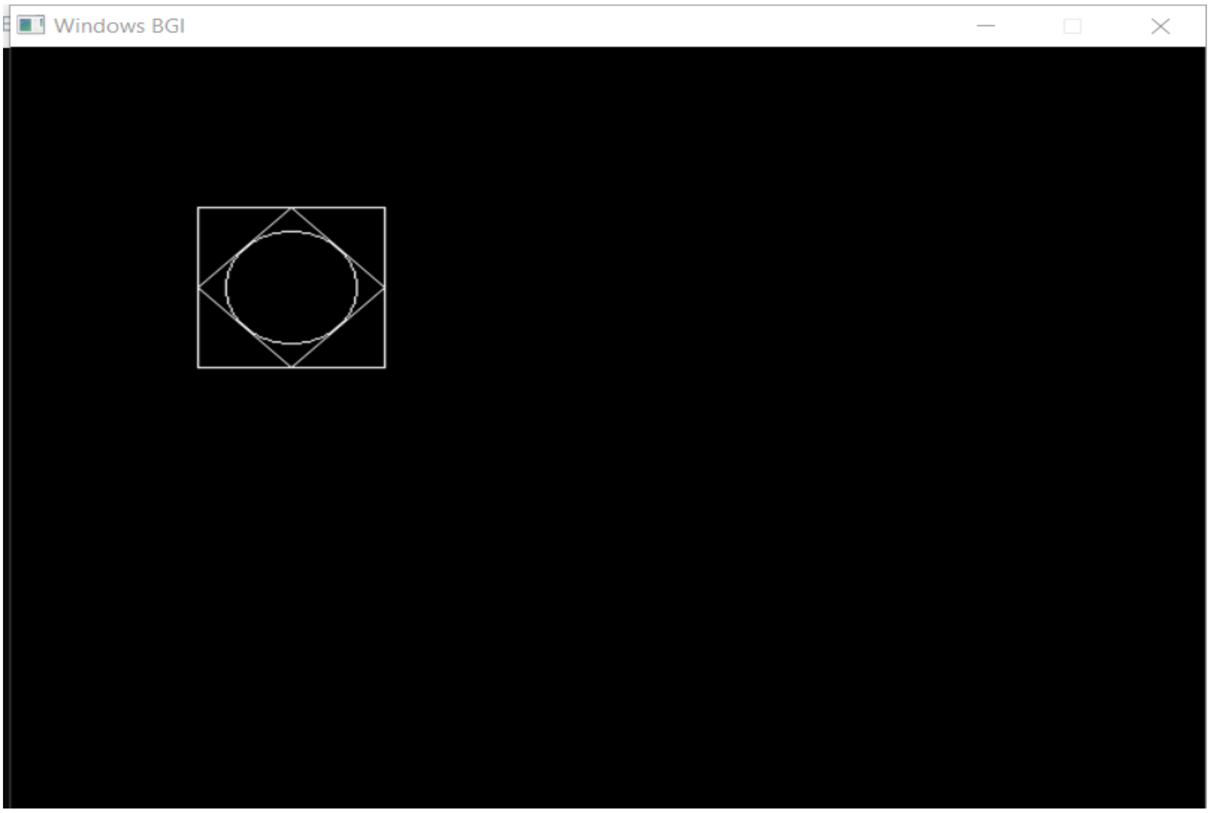
```
}
```

Output :-



"C:\Users\Pratham\#PICT\IMP DOCUMENTS(ACAD)\SEM 3\OOPCGL\CG Practicals\CG_Assign_6_b\bin\Debug\CG_Assign_6_b.exe"

```
Enter x1 :- 100
Enter y1 :- 100
Enter x2 :- 200
Enter y2 :- 200
Side 1 :- 100 Side 2 :- 100
Radius :- 35.3553
```



Windows BGI