# Assignment 4

**Code :-**

```
#include<windows.h>
#include<GL/glut.h>
#include<GL/glu.h>

int LE[500],RE[500];
int EdgeFill = 0,FillFlag = 0;

void Intersection(GLint x1,GLint y1,GLint x2,GLint y2)
{
   int y,t;
   float x,M;

   if(y2<y1)
   {
      t=x1;
      x1=x2;
      x2=t;

      t=y1;
      y1=y2;
```

```
    y2=t;
}

if(y1==y2)
{
    M=x2-x1;
}
else
{
   M=(x2-x1)/(y2-y1);
}

x=x1;
for(y=y1;y<=y2;y++)
{
    if(x<LE[y])
    {
        LE[y]=x;
    }
    if(x>RE[y])
    {
        RE[y]=x;
```

```
        }

        x = x+M;
    }

}

void Draw()
{

    glClear(GL_COLOR_BUFFER_BIT);
    //int LE[500],RE[500];
    int x,y,i;
    //GLint
P1[2]={125,250},P2[2]={250,125},P3[2]={375,250},P4[2]={250
,375};
    GLint
P1[2]={100,100},P2[2]={100,200},P3[2]={150,200},P4[2]={150
,150},P5[2]={200,150},P6[2]={200,100};

    for(i=0;i<500;i++)
    {
        LE[i]=500;
```

```
    RE[i]=0;
}

if(EdgeFill==1){
    glBegin(GL_LINE_LOOP);
        glVertex2iv(P1);
        glVertex2iv(P2);
        glVertex2iv(P3);
        glVertex2iv(P4);
        glVertex2iv(P5);
        glVertex2iv(P6);
        //glFlush();
    glEnd();
}

/*Intersection(P1[0],P1[1],P2[0],P2[1]);
Intersection(P2[0],P2[1],P3[0],P3[1]);
Intersection(P3[0],P3[1],P4[0],P4[1]);
Intersection(P4[0],P4[1],P1[0],P1[1]);*/
Intersection(P1[0],P1[1],P2[0],P2[1]);
Intersection(P2[0],P2[1],P3[0],P3[1]);
Intersection(P3[0],P3[1],P4[0],P4[1]);
```

```
    Intersection(P4[0],P4[1],P5[0],P5[1]);
    Intersection(P5[0],P5[1],P6[0],P6[1]);
    Intersection(P6[0],P6[1],P1[0],P1[1]);

    if(FillFlag == 1){
       for(y=0;y<500;y++)
       {
          for(x=LE[y];x<RE[y];x++)
          {
             glBegin(GL_POINTS);
               glVertex2i(x,y);
             glEnd();
             glFlush();
          }
       }
    }

    glFlush();

}

void Menu(int id)
```

```
{
   if(id == 1)
   {
      EdgeFill = 1;
   }
   else if(id == 2)
   {
      EdgeFill = 0;
   }
   else if(id == 3)
   {
      exit(0);
   }

   FillFlag = 1;
   glutPostRedisplay();
}

void MyInit()
{
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
```

```
    gluOrtho2D(0,500,0,500);
    glMatrixMode(GL_MODELVIEW);

    glutCreateMenu(Menu);
    glutAddMenuEntry("With Edge",1);
    glutAddMenuEntry("Without Edge",2);
    glutAddMenuEntry("Exit",3);
    glutAttachMenu(GLUT_RIGHT_BUTTON);
}


int main(int argC,char *argV[])
{
    glutInit(&argC,argV);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(500,500);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutCreateWindow("Scan FIll");
    MyInit();
    glutDisplayFunc(Draw);
    glutMainLoop();
```

```
    return 0;
}
```

**Output :-**