```
#include <iostream>
#include <math.h>
#include <time.h>
#include <GL/glut.h>
#include <vector>

using namespace std;

int edge;
vector<int> xpoint;
vector<int> ypoint;

int ch;

double round(double d){

   return floor(d + 0.5);
}

void init(){
   glClearColor(1.0,1.0,1.0,0.0);
   glMatrixMode(GL_PROJECTION);
   gluOrtho2D(0,640,0,480);
   glClear(GL_COLOR_BUFFER_BIT);
}



void rotaion(){
   int cx, cy;
   cout<<"\n Enter Ar point x , y ";
```

```
    cin >> cx >> cy;

    cx = cx+320;
    cy = cy+240;
    glColor3f(0.0, 1.0, 0.0);
    glBegin(GL_POINTS);
       glVertex2i(cx,cy);
    glEnd();
    glFlush();

    double the;
    cout<<"\n Enter thetha ";
    cin>>the;
    the = the * 3.14/180;

    glColor3f(0,0,1.0);
    glBegin(GL_POLYGON);
       for(int i=0;i<edge;i++){
          glVertex2i(round(((xpoint[i] - cx)*cos(the) - ((ypoint[i]-cy)*sin(the))) + cx),
                round(((xpoint[i] - cx)*sin(the) + ((ypoint[i]-cy)*cos(the))) + cy));
       }
    glEnd();
    glFlush();
}



void Draw(){



       glColor3f(1.0,0,0);
```

```
    glBegin(GL_LINES);
      glVertex2i(0,240);
      glVertex2i(640,240);
    glEnd();
    glColor3f(1.0,0,0);
    glBegin(GL_LINES);
      glVertex2i(320,0);
      glVertex2i(320,480);
    glEnd();
    glFlush();

    glColor3f(1.0,0,0);
    glBegin(GL_POLYGON);
      for(int i=0;i<edge;i++){
        glVertex2i(xpoint[i],ypoint[i]);
      }
    glEnd();
    glFlush();
  }

int main(int argc, char** argv){

    cout<<"Enter No of edges \n";
    cin>> edge;
```

```
int xpointnew, ypointnew;
cout<<" Enter"<< edge <<" point of polygon \n";
for(int i=0;i<edge;i++){

   cout<<"Enter "<< i << " Point ";
   cin>>xpointnew>>ypointnew;

   xpoint.push_back(xpointnew+320);
   ypoint.push_back(ypointnew+240);

}
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
   glutInitWindowSize(640,480);
   glutInitWindowPosition(200,200);
   glutCreateWindow("2D");
   init();
   glutDisplayFunc(Draw);
    rotaion();
glutMainLoop();
   return 0;
}
```