```
#define BLACK 0 #include <stdio.h>
#include <math.h> #include <GL/glut.h>
GLfloattriangle[3][3]={{100.0,250.0,175.0},{100.0,10
0.0,300.0},{1.0,1.0,1.0}};
GLfloat rotatemat[3][3]={{0},{0},{0}};
GLfloat result[3][3]={{0},{0},{0}};
GLfloat arbitrary_x=0; GLfloat arbitrary_y=0;
float rotation_angle;
void multiply()  { int i,j,k;
for(i=0;i<3;i++)   for(j=0;j<3;j++)  { result[i][j]=0;
for(k=0;k<3;k++)
result[i][j]=result[i][j]+rotatemat[i][k]*triangle[k][j]; }
}   void rotate()   { GLfloat m,n;
m=-arbitrary_x*(cos(rotation_angle) -1) + arbitrary_y
* (sin(rotation_angle));
n=-arbitrary_y * (cos(rotation_angle) - 1) -arbitrary_x
* (sin(rotation_angle));
rotatemat[0][0]=cos(rotation_angle);
rotatemat[0][1]=-sin(rotation_angle);
rotatemat[0][2]=m;rotatemat[1][0]=sin(rotation_angle)
; rotatemat[1][1]=cos(rotation_angle);
rotatemat[1][2]=n;  rotatemat[2][0]=0;
rotatemat[2][1]=0;  rotatemat[2][2]=1; multiply();   }
void drawtriangle()
{glColor3f(0.0, 0.0, 1.0);glBegin(GL_LINE_LOOP);
glVertex2f(triangle[0][0],triangle[1][0]);
glVertex2f(triangle[0][1],triangle[1][1]);
glVertex2f(triangle[0][2],triangle[1][2]);  glEnd();  }
void drawrotatedtriangle()
{ glColor3f(1.0, 0.0, 0.0);glBegin(GL_LINE_LOOP);
glVertex2f(result[0][0],result[1][0]);
glVertex2f(result[0][1],result[1][1]);
glVertex2f(result[0][2],result[1][2]);  glEnd();  }
void display() { glClear(GL_COLOR_BUFFER_BIT);
drawtriangle();drawrotatedtriangle();   glFlush(); }
void myinit()   { glClearColor(1.0,1.0,1.0,1.0);
glColor3f(1.0,0.0,0.0);  glPointSize(1.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();  gluOrtho2D(0.0,499.0,0.0,499.0); }
int main(int argc, char** argv)
{ int ch;printf("enter your choice \n1: Rotation about
origin \n2: Rotation about a Fixed point\n");
scanf("%d",&ch);
switch(ch)   {  case 1: printf("Enter the rotation angle
in degree :");  scanf("%f", &rotation_angle);
 rotation_angle= (3.14 * rotation_angle) / 180;
rotate();      break;
case 2: printf("Enter the fixed points :");
scanf("%f%f", &arbitrary_x,&arbitrary_y);
printf("Enter rotation angle in degree :");
scanf("%f", &rotation_angle);
rotation_angle= (3.14 * rotation_angle) / 180;
rotate();       break;   }
glutInit(&argc,argv);glutInitDisplayMode(GLUT_SIN
GLE|GLUT_RGB);glutInitWindowSize(500,500);
glutInitWindowPosition(0,0);glutCreateWindow("trian
gle rotation");glutDisplayFunc(display);
myinit();  glutMainLoop();  return 0;  }
```