Experiment 2:Create and rotate a triangle about the origin and a fixed point

AIM:

Write a program to Create and rotate a triangle about the origin and a fixed point.

PROGRAM:

```
#include<stdio.h>
#include<math.h>
#include<GL/glut.h>

GLfloatt[3][3]={{10.0,30.0,20.0},{20.0,20.0,40.0},{1.0,1.0,1.0}};
GLfloatrotatemat[3][3]={{0},{0},{0}};
GLfloatresult[3][9]={{0},{0},{0}};

GLfloatxr=10.0;
GLfloatyr=20.0;

GLfloat theta;
GLintch;

void multiply(){
        int i,j,k;
        for(i=0;i<3;i++)
                for(j=0;j<9;j++){
                        result[i][j]=0;
                        for(k=0;k<3;k++)
                                result[i][j]=result[i][j]+rotatemat[i][k]*t[k][j];
                }
}

void rotate_about_origin(){
        rotatemat[0][0]=cos(theta);
        rotatemat[0][1]=-sin(theta);
        rotatemat[0][2]=0;
        rotatemat[1][0]=sin(theta);
        rotatemat[1][1]=cos(theta);
        rotatemat[1][2]=0;
        rotatemat[2][0]=0;
        rotatemat[2][1]=0;
        rotatemat[2][2]=1;
        multiply();
}
```

```
void rotate_about_fixed_point(){
	GLfloatm,n;
	m=xr*(1-cos(theta))+yr*sin(theta);
	n=yr*(1-cos(theta))-xr*sin(theta);
	rotatemat[0][0]=cos(theta);
	rotatemat[0][1]=-sin(theta);
	rotatemat[0][2]=m;
	rotatemat[1][0]=sin(theta);
	rotatemat[1][1]=cos(theta);
	rotatemat[1][2]=n;
	rotatemat[2][0]=0;
	rotatemat[2][1]=0;
	rotatemat[2][2]=1;
	multiply();
}

void draw_triangle(){
	glLineWidth(10);
	glBegin(GL_LINE_LOOP);
	glColor3f(1.0,0.0,0.0);
	glVertex2f(t[0][0],t[1][0]);
	glColor3f(0.0,1.0,0.0);
	glVertex2f(t[0][1],t[1][1]);
	glColor3f(0.0,0.0,1.0);
	glVertex2f(t[0][2],t[1][2]);
	glEnd();
	glFlush();
}

void draw_rotated_triangle(){
	glLineWidth(10);
	glBegin(GL_LINE_LOOP);
	glColor3f(1.0,0.0,0.0);
	glVertex2f(result[0][0],result[1][0]);
	glColor3f(0.0,1.0,0.0);
	glVertex2f(result[0][1],result[1][1]);
	glColor3f(0.0,0.0,1.0);
	glVertex2f(result[0][2],result[1][2]);
	glEnd();
	glFlush();
}

void display(){
	glClear(GL_COLOR_BUFFER_BIT);
	if(ch==1){
		draw_triangle();
```

```
                rotate_about_origin();
                draw_rotated_triangle();
                glFlush();
        }
        if(ch==2){
                draw_triangle();
                rotate_about_fixed_point();
                draw_rotated_triangle();
                glFlush();
        }
}

void myinit(){
        glClearColor(1.0,1.0,1.0,1.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(-50.0,50.0,-50.0,50.0);
}

int main(int argc,char** argv){
        printf("***Rotation***\n1.Rotation about origin\n2.Rotation about a fixed point
(xr,yr)\n");
        printf("Enter choice\n");
        scanf("%d",&ch);
        printf("Enter the rotation angle\n");
        scanf("%f",&theta);
        theta=theta*(3.14/180);
        glutInit(&argc,argv);
        glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
        glutInitWindowSize(500,500);
        glutInitWindowPosition(0,0);
        glutCreateWindow("Triangle Rotation\n");
        glutDisplayFunc(display);
        myinit();
        glutMainLoop();
        return 0;
}
```

Sample Output :