**NAME: Shivani M V  
 USN: 1CR18CS152  
 BATCH: C1**

**TRIANGLE ROTATION**

**AIM:** Create and rotate a triangle about the origin and a fixed point.

**CODE:**

#include<GL/gl.h>

#include<GL/glu.h>

#include<GL/glut.h>

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

float triangle[3][3] = {{150,250,350},{200,400,200},{1,1,1}};

float rot\_mat[3][3] = {{0},{0},{0}};

float result[3][3] = {{0},{0},{0}};

float h=0,k=0;

float m=0,n=0;

float theta;

void drawTriangle(float t[3][3]){

glBegin(GL\_TRIANGLES);

glVertex2f(t[0][0],t[1][0]);

glVertex2f(t[0][1],t[1][1]);

glVertex2f(t[0][2],t[1][2]);

glEnd();

}

void multiply(){

int i,j,l;

for(i=0;i<3;i++){

for(j=0;j<3;j++){

result[i][j]=0;

for(l=0;l<3;l++)

result[i][j]=result[i][j]+rot\_mat[i][l]\*triangle[l][j];

}

}

}

void rrotate(float m,float n)

{

rot\_mat[0][0] = cos(theta);

rot\_mat[0][1] = -sin(theta);

rot\_mat[0][2] = m;

rot\_mat[1][0] = sin(theta);

rot\_mat[1][1] = cos(theta);

rot\_mat[1][2] = n;

rot\_mat[2][0] = 0;

rot\_mat[2][1] = 0;

rot\_mat[2][2] = 1;

multiply();

}

void display() {

glClearColor(1,1,1,1);

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0,0,1);

drawTriangle(triangle);

glColor3f(1,0,0);

rrotate(0,0);

drawTriangle(result);

m = -h\*(cos(theta)-1)+k\*(sin(theta));

n = -k\*(cos(theta)-1)-h\*(sin(theta));

rrotate(m,n);

printf("%f , %f",result[0][0],result[1][0]);

glColor3f(0,1,0);

drawTriangle(result);

glFlush();

}

void init()

{

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(-500,500,-500,500);

glMatrixMode(GL\_MODELVIEW);

}

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

{

printf("Enter the Rotation angle \n");

scanf("%f",&theta);

printf("Enter the two fixed points \n");

scanf("%f %f",&h,&k);

theta = ((3.14/180)\*theta);

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(800,600);

glutInitWindowPosition(100,100);

glutCreateWindow("TRIANGLE");

init();

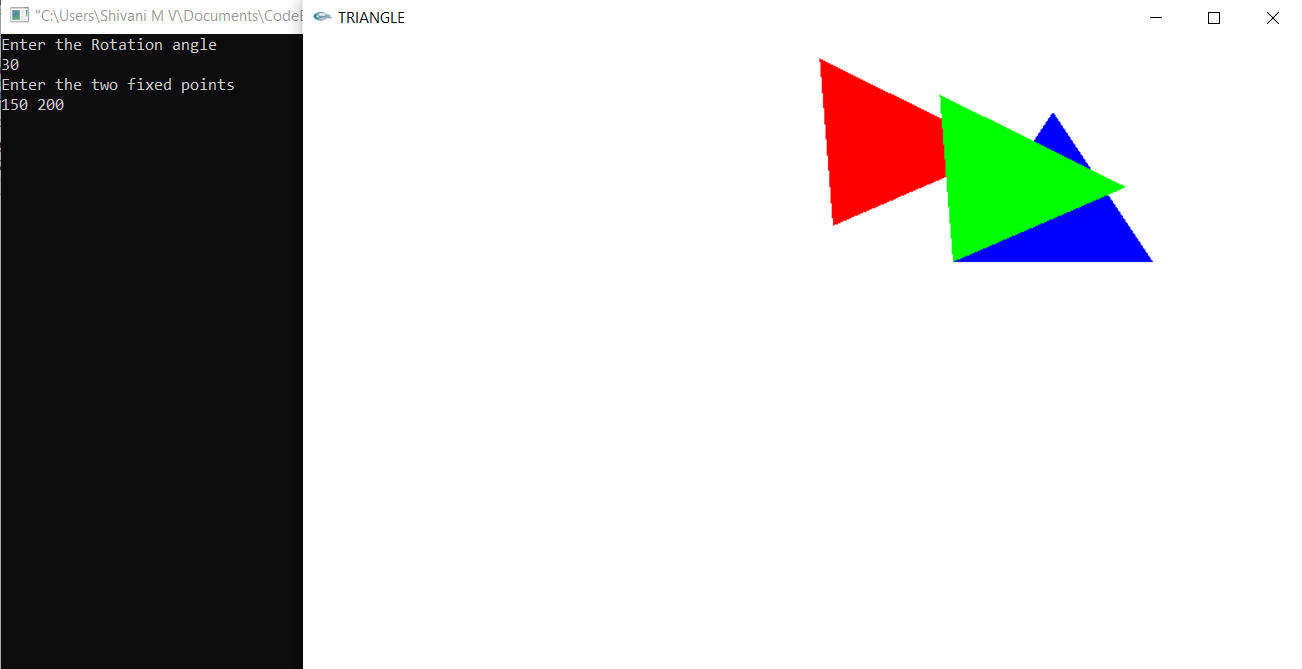
glutDisplayFunc(display);

glutMainLoop();

}

**OUTPUT:**

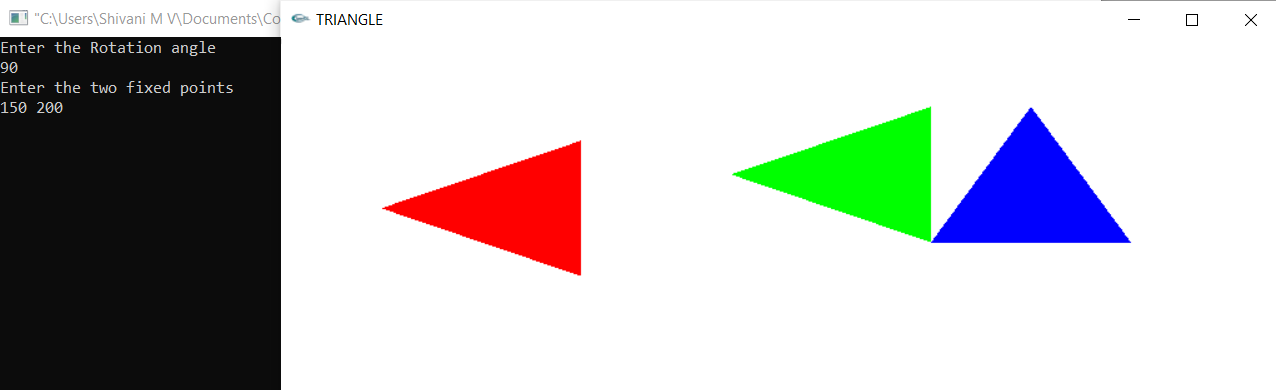
**Theta = 30**



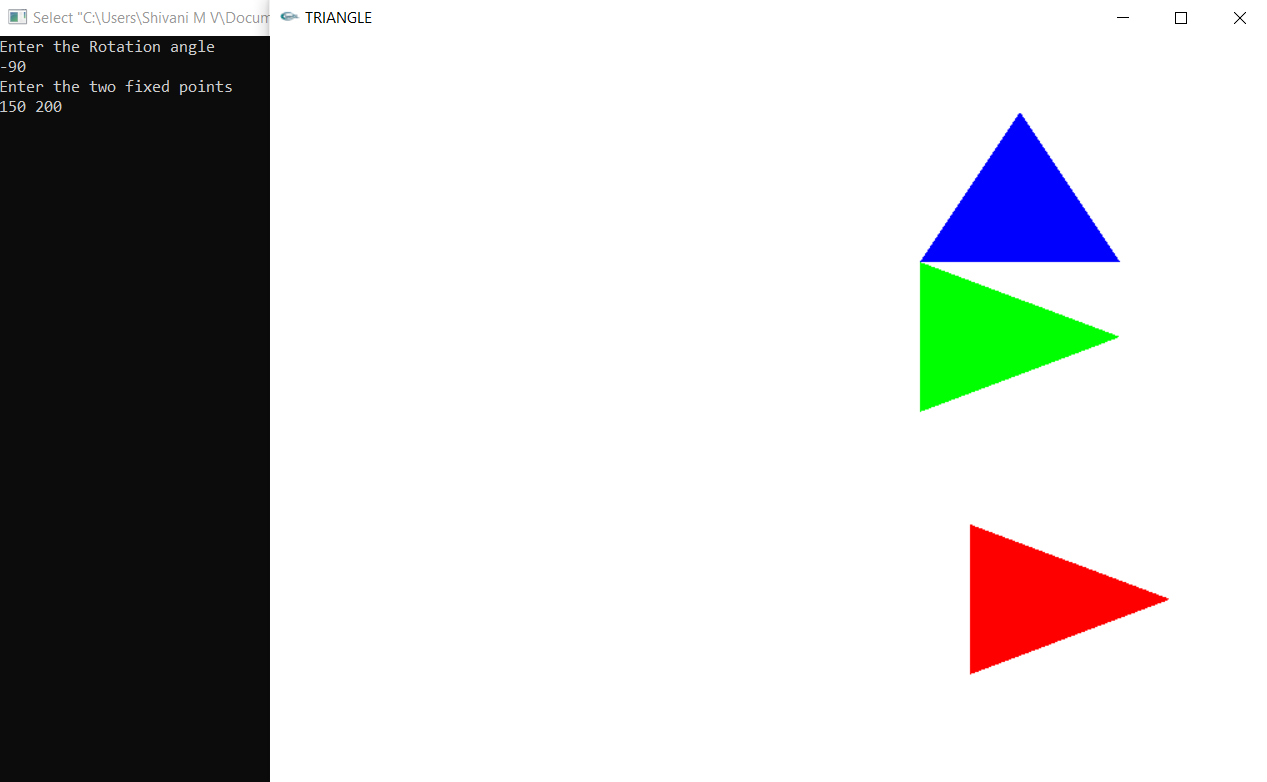
**Theta = -30**



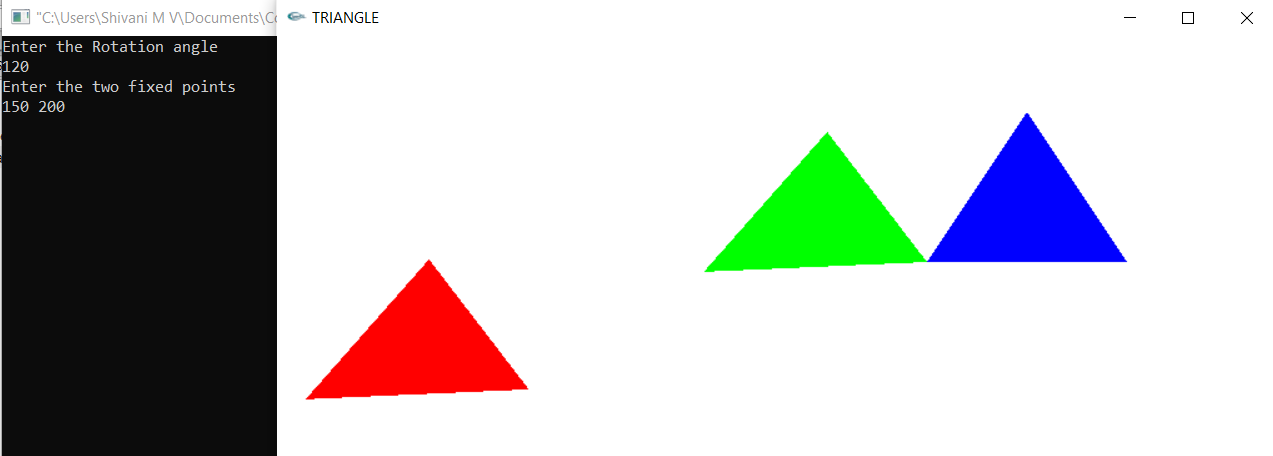
**Theta = 90**



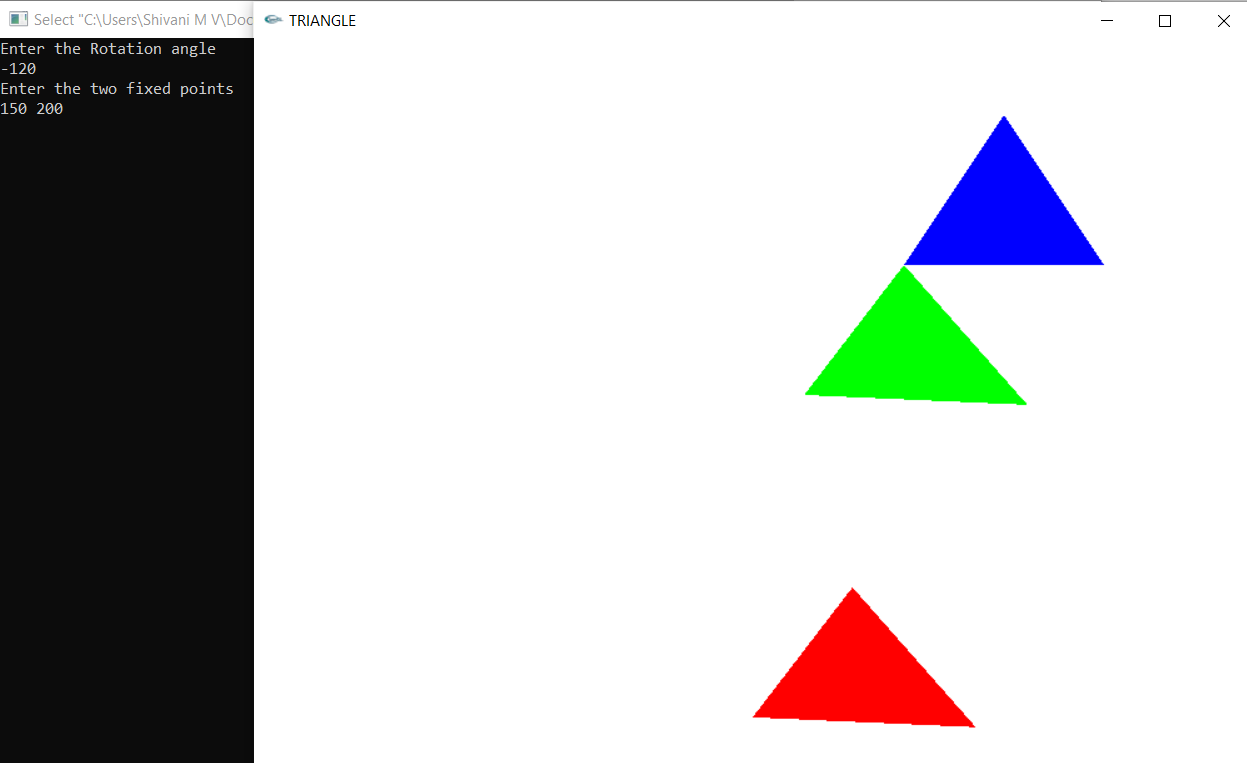
**Theta = -90**



**Theta = 120**



**Theta = -120**



**Theta = 180**

**Around centroid = (250,266.66)**

