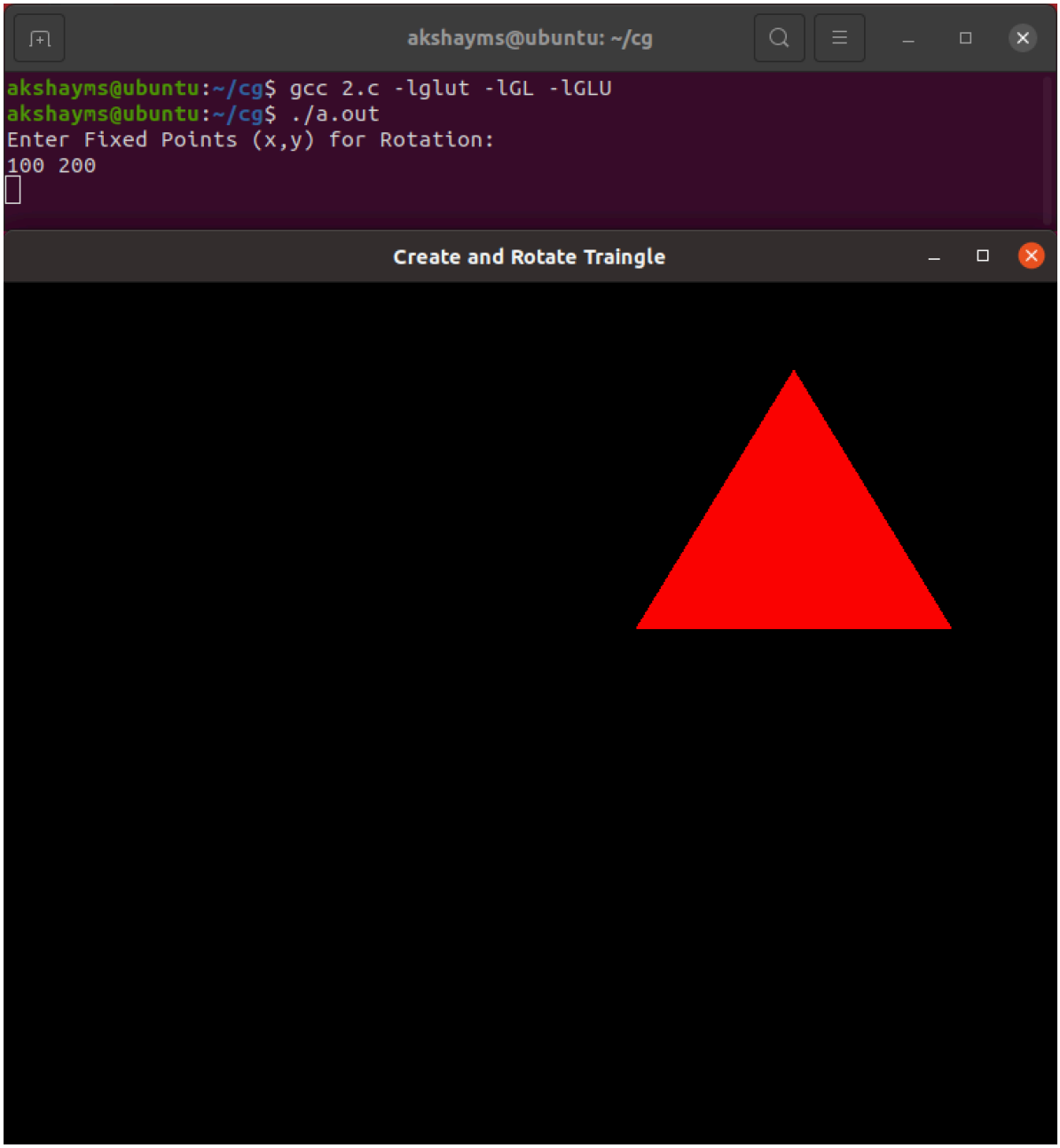
2. Create and Rotate a triangle about the origin and a fixed point

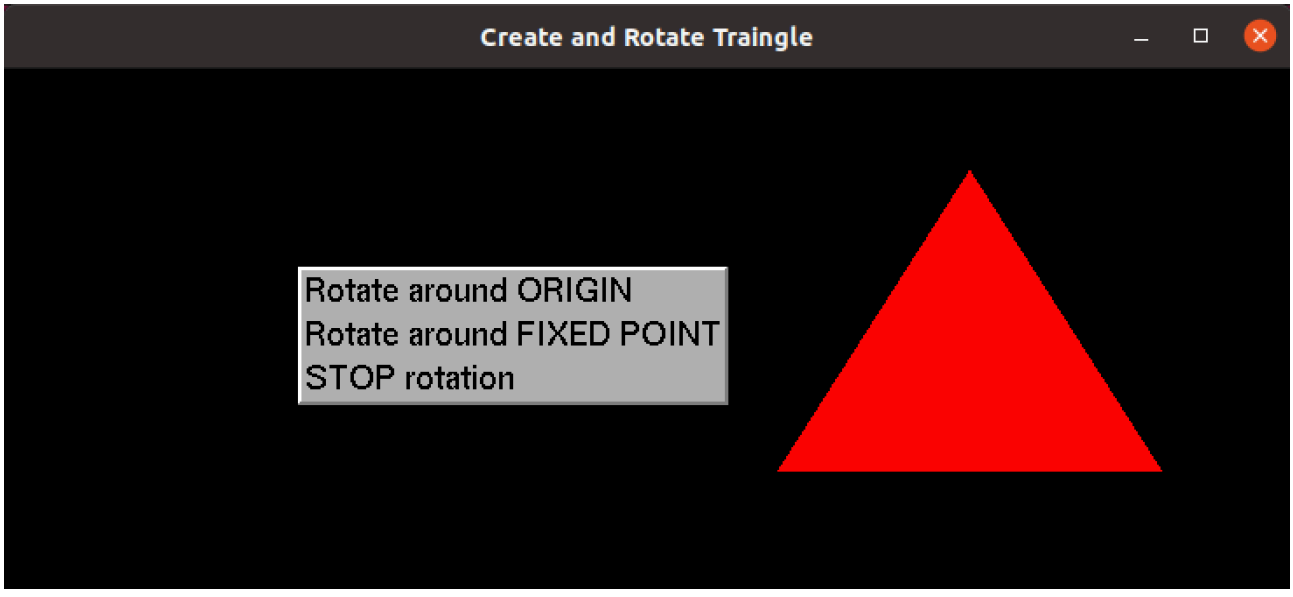
```
#include<GL/glut.h>
#include<stdio.h>
int x,y;
int rFlag=0;

void draw_pixel(float x1, float y1){
	glColor3f(0.0,0.0,1.0);
	glPointSize(5.0);
	glBegin(GL_POINTS);
	glVertex2f(x1,y1);
	glEnd();
}

void triangle(){
	glColor3f(1.0,0.0,0.0);
	glBegin(GL_POLYGON);
	glVertex2f(100,100);
	glVertex2f(250, 400);
	glVertex2f(400,100);
	glEnd();
}

float th=0.0;
float trX = 0.0, trY = 0.0;

void display(){
	glClear(GL_COLOR_BUFFER_BIT);
	glLoadIdentity();

	if(rFlag==1){
		trX=0.0;
		trY=0.0;
		th+=1;
		draw_pixel(0.0,0.0);
	}
	if(rFlag==2){
		trX=x;
		trY=y;
		th+=1;
		draw_pixel(x,y);
	}
	glTranslatef(trX,trY,0.0);
	glRotatef(th,0.0,0.0,1.0);
	glTranslatef(-trX,-trY,0.0);
	triangle();
```

```
	glutPostRedisplay();
	glutSwapBuffers();
}

void myInit(){
	glClearColor(0.0,0.0,0.0,1.0);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(-500.0,500.0,-500.0,500.0);
	glMatrixMode(GL_MODELVIEW);
}

void rotateMenu(int option){
	if(option==1)
		rFlag=1;
	if(option==2)
		rFlag=2;
	if(option==3)
		rFlag=3;
}

int main(int argc,char** argv){
	printf("Enter Fixed Points (x,y) for Rotation: \n");
	scanf("%d%d",&x,&y);
	glutInit(&argc,argv);
	glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
	glutInitWindowSize(500,500);
	glutInitWindowPosition(0,0);
	glutCreateWindow("Create and Rotate Traingle");
	myInit();
	glutDisplayFunc(display);
	glutCreateMenu(rotateMenu);
	glutAddMenuEntry("Rotate around ORIGIN",1);
	glutAddMenuEntry("Rotate around FIXED POINT",2);
	glutAddMenuEntry("STOP rotation",3);
	glutAttachMenu(GLUT_RIGHT_BUTTON);
	glutMainLoop();
}
```

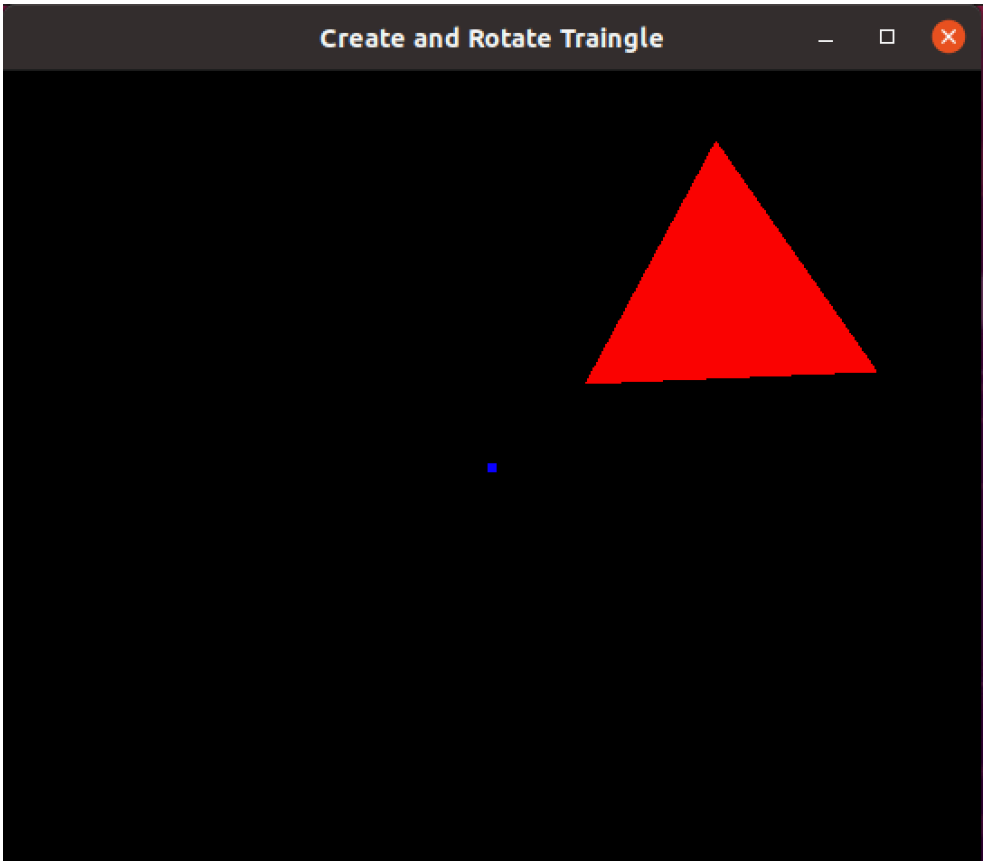
Output


akshayms@ubuntu: ~/cg
akshayms@ubuntu:~/cg$ gcc 2.c -lglut -lGL -lGLU
akshayms@ubuntu:~/cg$ ./a.out
Enter Fixed Points (x,y) for Rotation:
100 200
Create and Rotate Traingle

Menu


Create and Rotate Traingle
Rotate around ORIGIN
Rotate around FIXED POINT
STOP rotation

Rotation around ORIGIN



Rotation around FIXED POINT