5. Write a program to create a house like figure and perform the following operations.

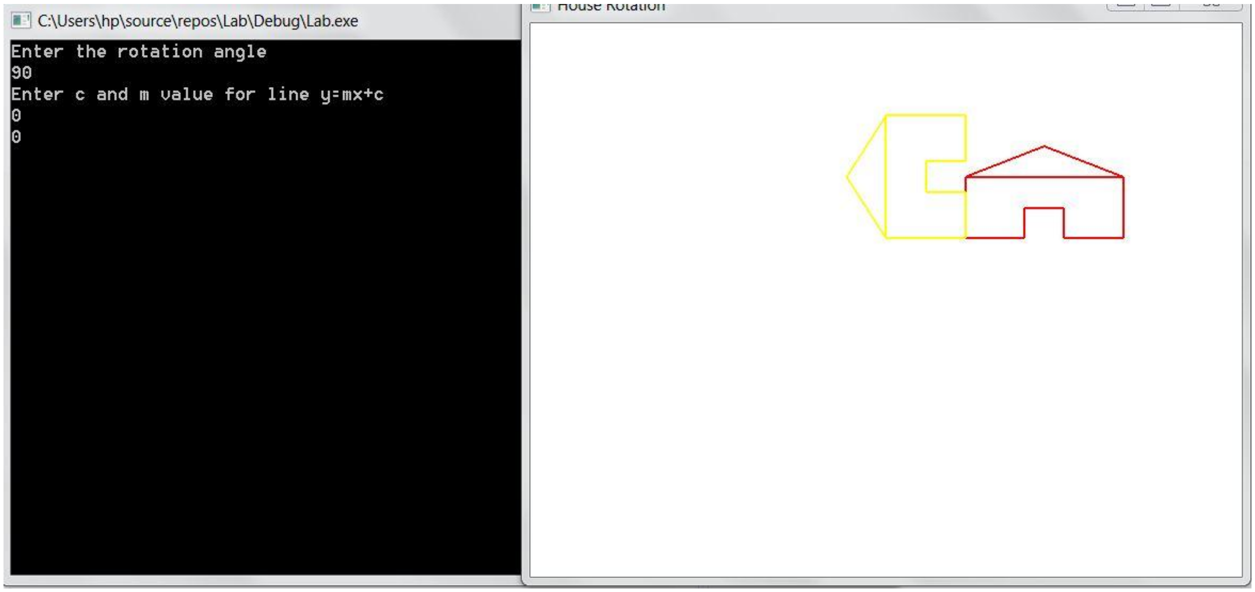
i. Rotate it about a given fixed point using OpenGL transformation functions.

ii. Reflect it about an axis y=mx+c using OpenGL transformation functions.

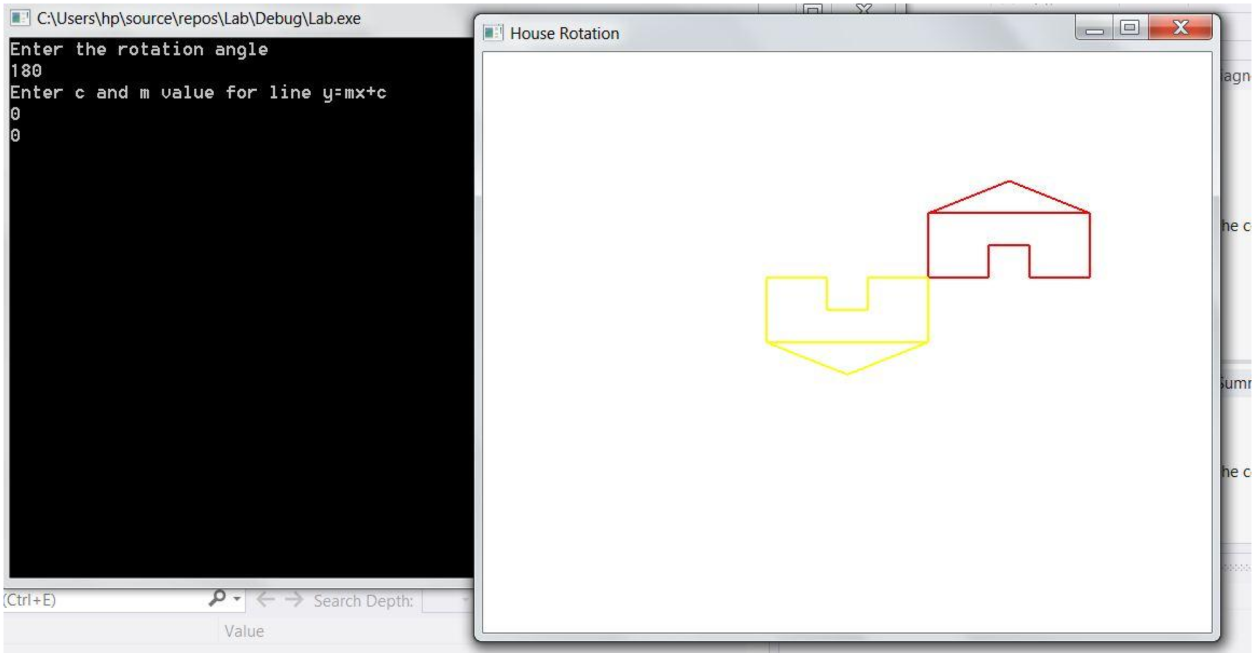
```
#include<gl/glut.h>
#include <math.h>
#include<stdlib.h>
#include<stdio.h>

//RIGHT CLICK TO SHOW REFLECTED HOUSE

float house[11][2] = { { 100,200 },{ 200,250 },{ 300,200 },{ 100,200 },{ 100,100 },{ 175,100
},{ 175,150 },{ 225,150 },{ 225,100 },{ 300,100 },{ 300,200 } };
int angle;
float m, c, theta;
void display()
{
	glClearColor(1, 1, 1, 0);
	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(-450, 450, -450, 450);
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	//NORMAL HOUSE
	glColor3f(1, 0, 0);
	glBegin(GL_LINE_LOOP);
	for (int i = 0; i < 11; i++)
		glVertex2fv(house[i]);
	glEnd();
	glFlush();
	//ROTATED HOUSE
	glPushMatrix();
	glTranslatef(100, 100, 0);
	glRotatef(angle, 0, 0, 1);
	glTranslatef(-100, -100, 0);
	glColor3f(1, 1, 0);
	glBegin(GL_LINE_LOOP);
	for (int i = 0; i < 11; i++)
		glVertex2fv(house[i]);
	glEnd();
	glPopMatrix();
```

```
        glFlush();
}
void display2()
{
        glClearColor(1, 1, 1, 0);
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(-450, 450, -450, 450);
        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();
        //normal house
        glColor3f(1, 0, 0);
        glBegin(GL_LINE_LOOP);
        for (int i = 0; i < 11; i++)
                glVertex2fv(house[i]);
        glEnd();
        glFlush();
        // line
        float x1 = 0, x2 = 500;
        float y1 = m * x1 + c;
        float y2 = m * x2 + c;
        glColor3f(1, 1, 0);
        glBegin(GL_LINES);
        glVertex2f(x1, y1);
        glVertex2f(x2, y2);
        glEnd();
        glFlush();

        //Reflected
        glPushMatrix();
        glTranslatef(0, c, 0);
        theta = atan(m);
        theta = theta * 180 / 3.14;
        glRotatef(theta, 0, 0, 1);
        glScalef(1, -1, 1);
        glRotatef(-theta, 0, 0, 1);
        glTranslatef(0, -c, 0);
        glBegin(GL_LINE_LOOP);
        for (int i = 0; i < 11; i++)
                glVertex2fv(house[i]);
        glEnd();
        glPopMatrix();
        glFlush();
}
void myInit() {
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(1.0, 0.0, 0.0);
        glLineWidth(2.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(-450, 450, -450, 450);
}
void mouse(int btn, int state, int x, int y) {
        if (btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
                display();
        }
        else if (btn == GLUT_RIGHT_BUTTON && state == GLUT_DOWN) {
                display2();
        }
}
void main(int argc, char** argv)
{
        printf("Enter the rotation angle\n");
        scanf_s("%d", &angle);
        printf("Enter c and m value for line y=mx+c\n");
        scanf_s("%f %f", &c, &m);
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(900, 900);
        glutInitWindowPosition(100, 100);
        glutCreateWindow("House Rotation");
        glutDisplayFunc(display);
        glutMouseFunc(mouse);
        myInit();
        glutMainLoop();
}
```

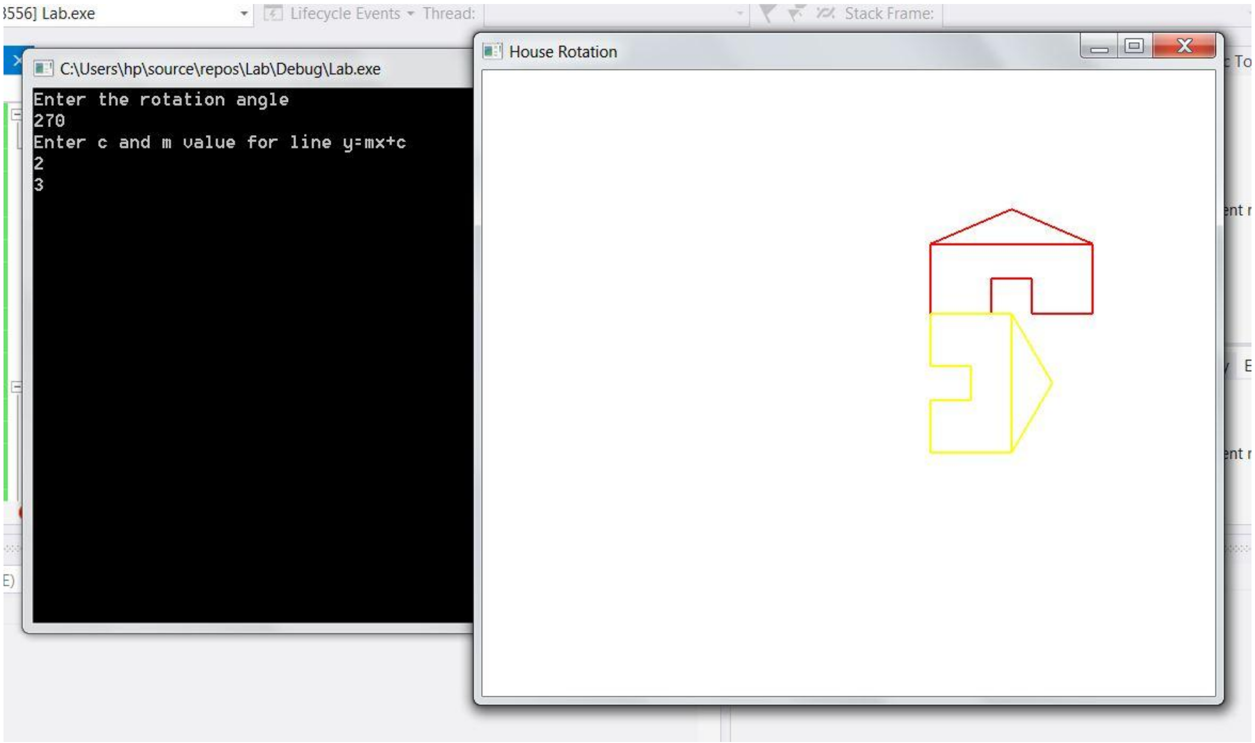
Output:-



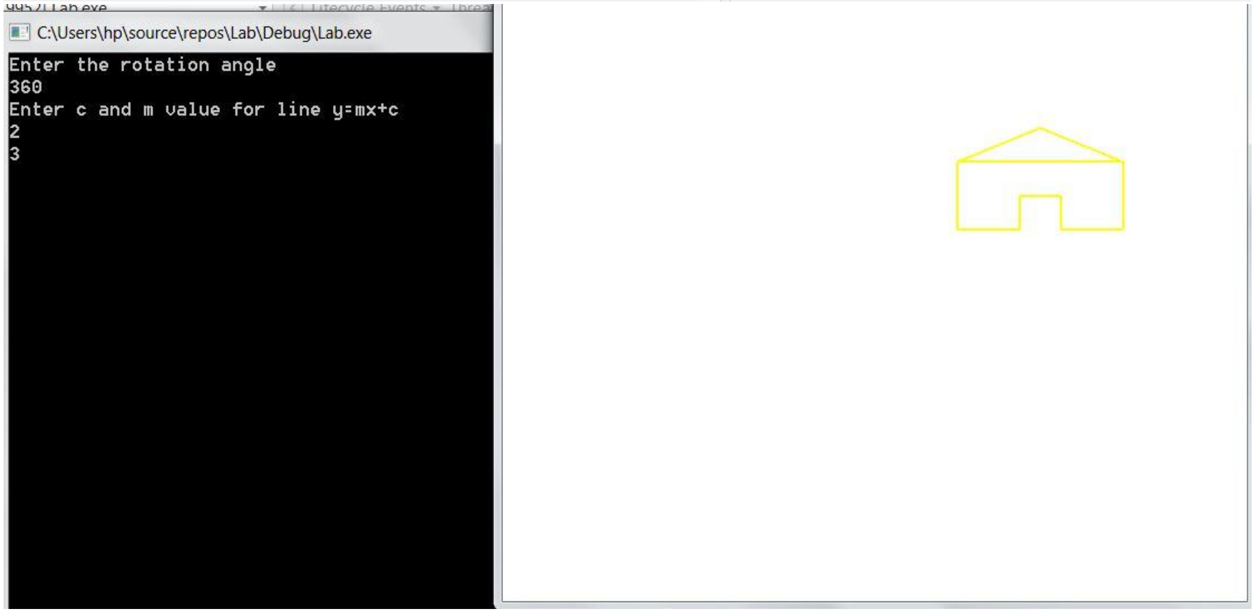
```
Enter the rotation angle
90
Enter c and m value for line y=mx+c
0
0
```



```
Enter the rotation angle
180
Enter c and m value for line y=mx+c
0
0
```



```
Enter the rotation angle
270
Enter c and m value for line y=mx+c
2
3
```



```
Enter the rotation angle
360
Enter c and m value for line y=mx+c
2
3
```