```c
#include<GL/glut.h>
#include<stdio.h>
int x,y;
int where_to_rotate=0;      // don't rotate initially
float rotate_angle=0;          // initial angle
float translate_x=0,translate_y=0;  // initial translation

void draw_pixel(float x1, float y1)
{
    glPointSize(5);
    glBegin(GL_POINTS);
        glVertex2f(x1,y1);      // plot a single point
    glEnd();
}

void triangle(int x, int y)
{
    glColor3f(1,0,0);
    glBegin(GL_POLYGON);    // drawing a Triangle
        glVertex2f(x,y);
        glVertex2f(x+400,y+300);
        glVertex2f(x+300,y+0);
    glEnd();
}

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

    glColor3f(1,1,1);               // mark origin point as white dot
    draw_pixel(0,0);              // plot origin - white colour

    if (where_to_rotate == 1) //Rotate Around origin
    {
        translate_x = 0;            // no translation for rotation around origin
        translate_y = 0;
        rotate_angle += 1;        // the amount of rotation angle
    }

    if (where_to_rotate == 2) //Rotate Around Fixed Point
    {
        translate_x = x;         // SET the translation to wherever the customer says
        translate_y = y;
        rotate_angle += 1;     // the amount of rotation angle
        glColor3f(0,0,1);        // mark the customer coordinate as blue dot
        draw_pixel(x,y);         // plot the customer coordinate - blue colour
    }

    glTranslatef(translate_x, translate_y, 0);    // ACTUAL translation +ve
    glRotatef(rotate_angle, 0, 0, 1);                        // rotate
    glTranslatef(-translate_x, -translate_y, 0);  // ACTUAL translation -ve

    triangle(translate_x,translate_y);                  // what to rotate? - TRIANGLE

    glutPostRedisplay();                  // call display function again and again
    glutSwapBuffers();                   // show the output
}

void init()
{
    glClearColor(0,0,0,1); //setting to black
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-800, 800, -800, 800);
    glMatrixMode(GL_MODELVIEW);
}

void rotateMenu (int option)
{
    if(option==1)
        where_to_rotate=1;      // rotate around origin

    if(option==2)
        where_to_rotate=2;      // rotate around customer's coordinates

    if(option==3)
        where_to_rotate=3;      // stop rotation
}

int main(int argc, char **argv)
{
    printf( "Enter Fixed Points (x,y) for Rotation: \n");
    scanf("%d %d", &x, &y);                 // getting the user's coordinates to rotate
```

```
85
86      glutInit(&argc, argv);                      // initialize the graphics system
87      glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);  // SINGLE also works
88      glutInitWindowSize(800, 800);    // 800 by 800 size..you can change it
89      glutInitWindowPosition(0, 0);       // where do you wanna see your window
90      glutCreateWindow("Create and Rotate Triangle");     // title
91
92      init();                                                 // initialize the canvas
93
94      glutDisplayFunc(display);            // call display function
95
96      glutCreateMenu(rotateMenu);     // menu items
97      glutAddMenuEntry("Rotate around ORIGIN",1);
98      glutAddMenuEntry("Rotate around FIXED POINT",2);
99      glutAddMenuEntry("Stop Rotation",3);
100     glutAttachMenu(GLUT_RIGHT_BUTTON);
101
102     glutMainLoop();                             // run forever
103 }
104
```