Lab Program-4

Create and rotate a triangle about the origin and a fixed point

```
/*Create and rotate a triangle about the origin and a fixed point */
#include<GL/glut.h>
#include<stdio.h>
int x,y;
int where_to_rotate;
float rotate_angle;
float translate_x, translate_y;
void draw_pixel(float x1,float y1)
{
   glPointSize(10.0);
   glBegin(GL_POINTS);
   glVertex2f(x1,y1);
    glEnd( );
   }
```

```
void triangle(int x, int y)
{
   glColor3f(0.0,1.0,0.0);
   glBegin(GL_TRIANGLES);
   glVertex2f(x,y);
   glVertex2f(x+400,y+300);
   glVertex2f(x+300,y+10);
   glEnd();
}
void display()
{
   glClear(GL_COLOR_BUFFER_BIT);
   glLoadIdentity();
   glColor3f(0,0,0);
   draw_pixel(0.0,0.0);
```

(OR)

```
glColor3f(0.0,1.0,0.0);
glBegin(GL_TRIANGLES);
glVertex2f(100,100);
glVertex2f(250,400);
glVertex2f(400,100);
glEnd();
```

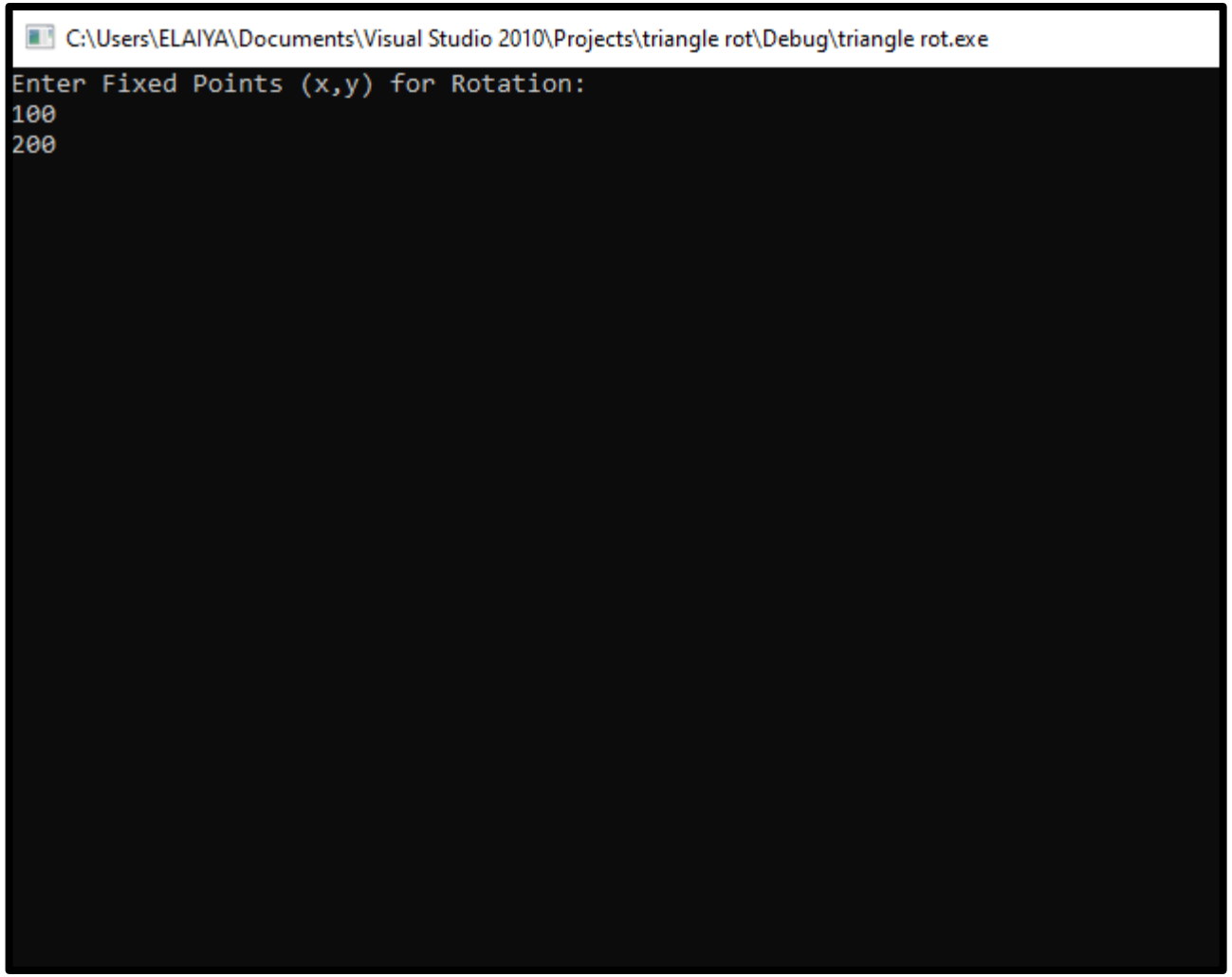
```
if(where_to_rotate==1) //Rotate About origin
  {
     translate_x=0.0;
          translate_y=0.0;
                    rotate_angle+=.3;
  }
  if(where_to_rotate==2) //Rotate About Fixed Point
  {
     translate_x=x;
                    translate_y=y;
                    rotate_angle+=.3;
          glColor3f(1.0,0.0,0.0);
          draw_pixel(x,y);
  }
```

```
glTranslatef(translate_x,translate_y, 0.0);
glRotatef(rotate_angle,0.0,0.0,1.0);
glTranslatef(-translate_x,-translate_y,0.0);
triangle(translate_x,translate_y);
glutPostRedisplay();
glutSwapBuffers();
 }
void myInit()
{
   glClearColor(1.0,1.0,1.0,1.0);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(-800.0, 800.0, -800.0, 800.0);
   glMatrixMode(GL_MODELVIEW);
}
```

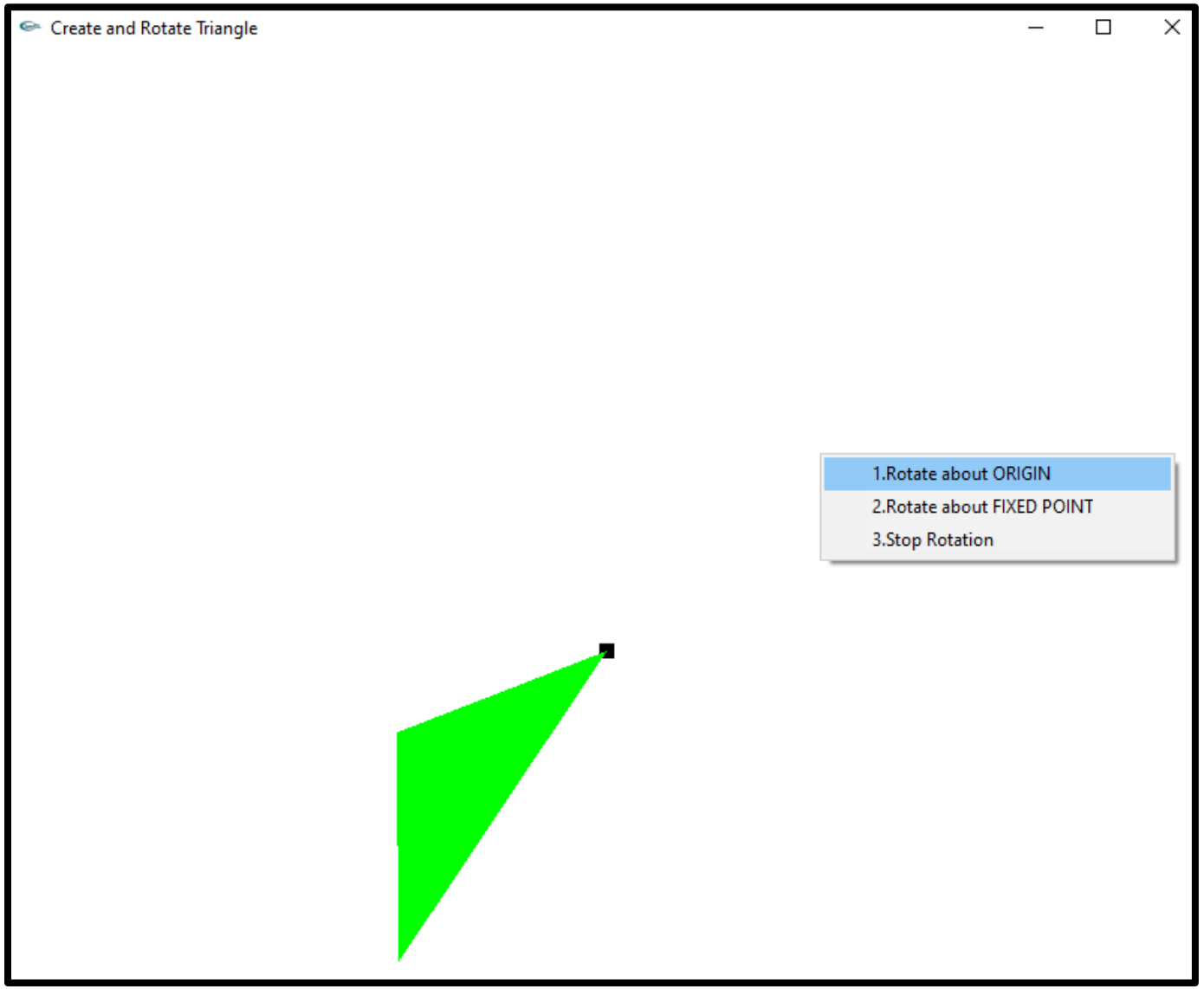
```
void rotateMenu (int option)
{
   if(option==1)
      where_to_rotate=1;
         if(option==2)
      where_to_rotate=2;
      if(option==3)
      where_to_rotate=3;
 }
```

```
   void 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_RGBA|GLUT_DEPTH);
   glutInitWindowSize(800, 800);
   glutInitWindowPosition(0, 0);
   glutCreateWindow("Create and Rotate Triangle");
   myInit();
   glutDisplayFunc(display);
   glutCreateMenu(rotateMenu);
   glutAddMenuEntry("1.Rotate about ORIGIN",1);
   glutAddMenuEntry("2.Rotate about FIXED POINT",2);
   glutAddMenuEntry("3.Stop Rotation",3);
   glutAttachMenu(GLUT_RIGHT_BUTTON);
   glutMainLoop();
}
```

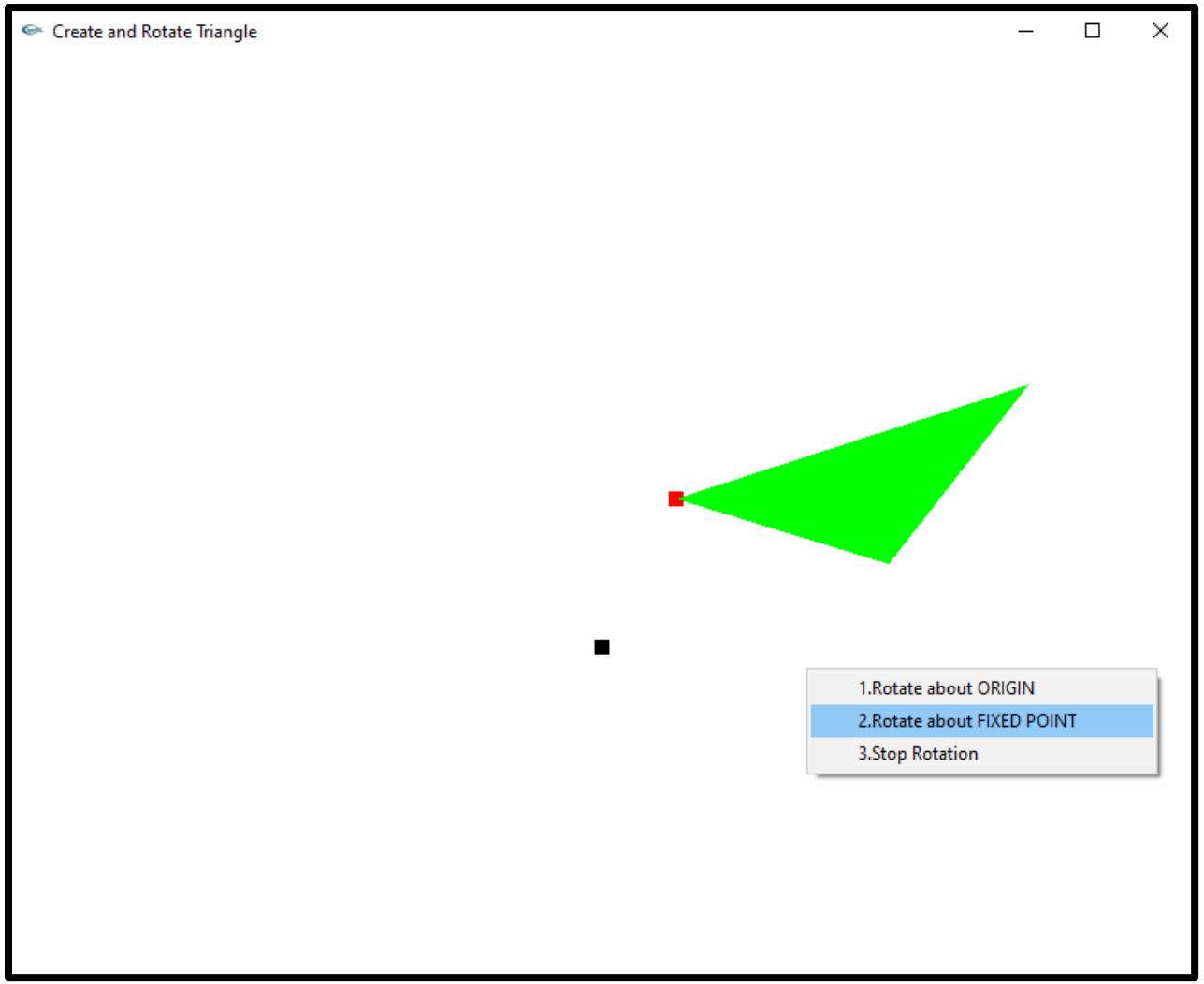
OUTPUT: ENTER THE X AND Y VALUE 100,200



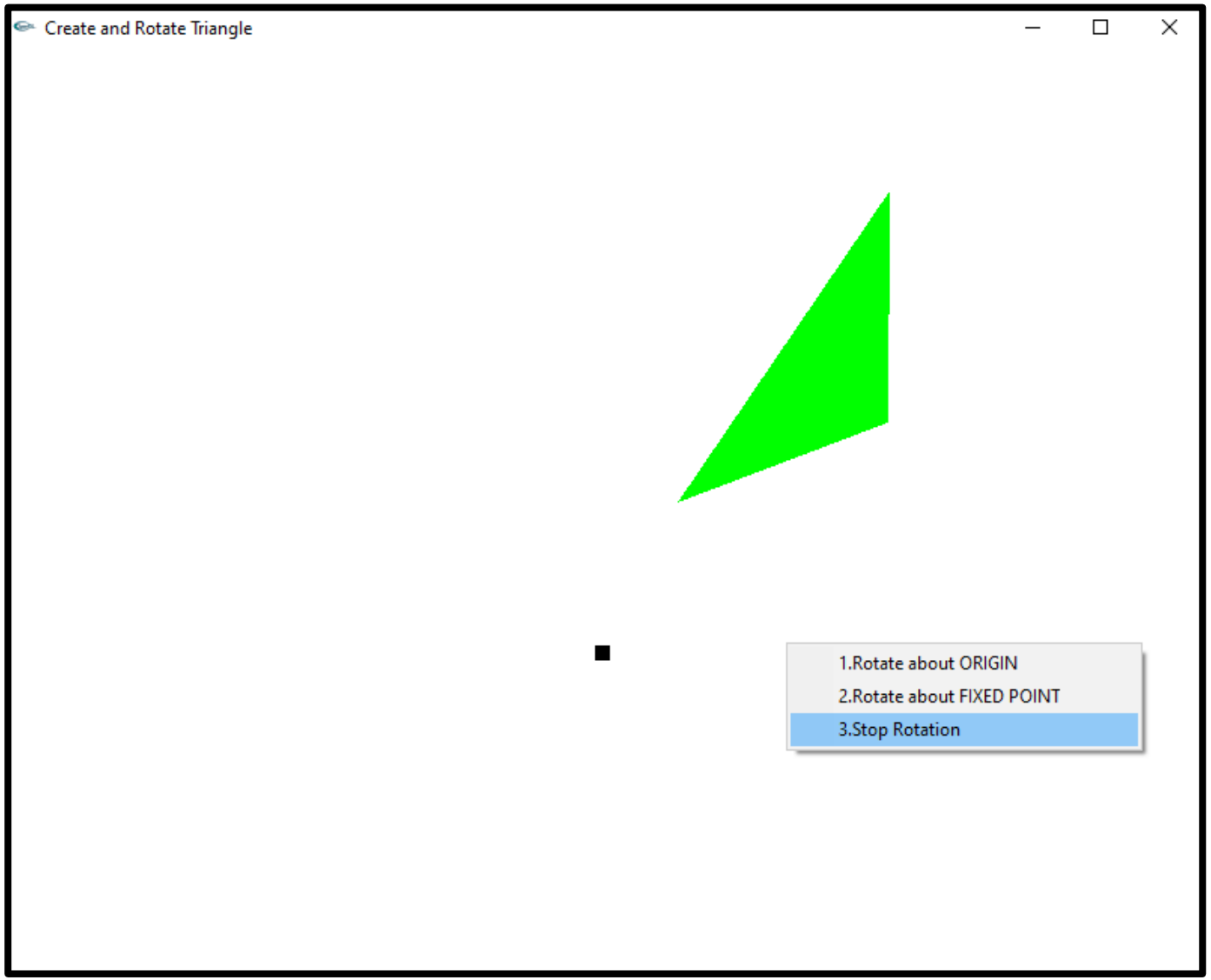
OUTPUT: 1.Rotate about ORIGIN



OUTPUT: 2.Rotate about FIXED POINT

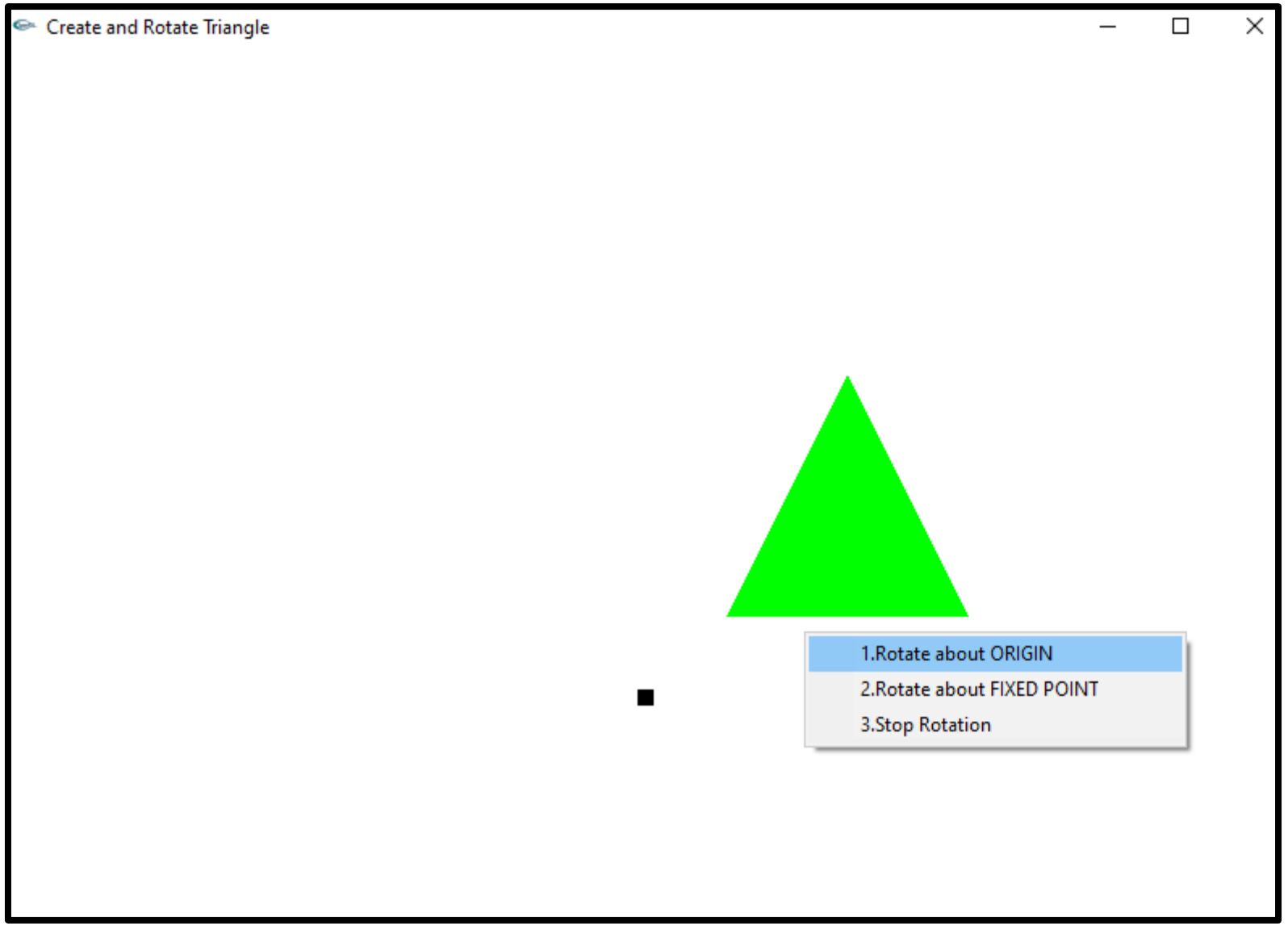


OUTPUT: 3.Stop Rotation

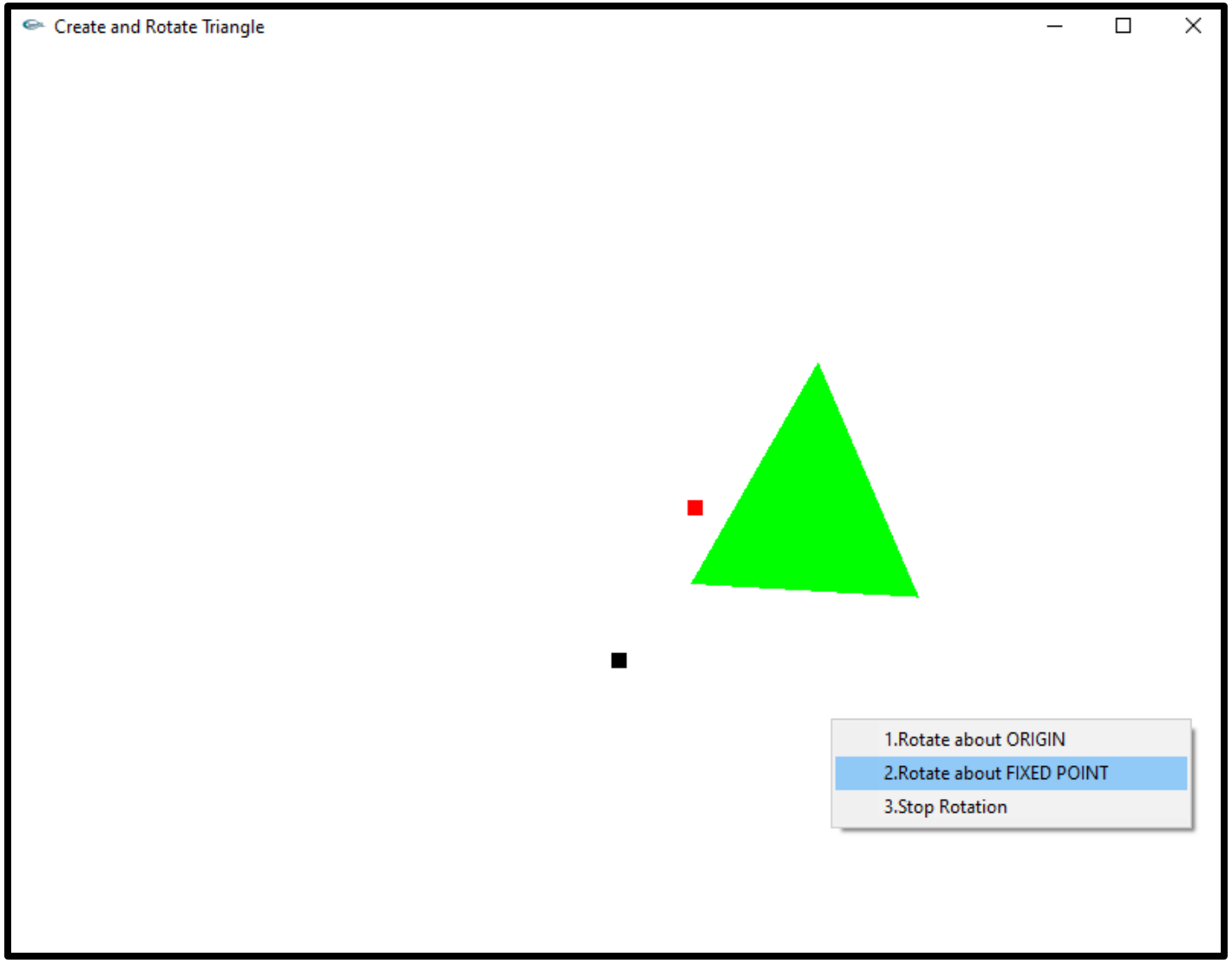


```
glColor3f(0.0,1.0,0.0);
glBegin(GL_TRIANGLES);
glVertex2f(100,100);
glVertex2f(250,400);
glVertex2f(400,100);
glEnd();
```

OUTPUT: 1.Rotate about ORIGIN



OUTPUT: 2.Rotate about FIXED POINT



OUTPUT: 3.Stop Rotation