RAMAIAH UNIVERSITY
OF APPLIED SCIENCES

ASSIGNMENT

Course Code	CSC309A
Course Name	Computer Graphics
Programme	B.Tech
Department	Computer Science and Engineering
Faculty	Faculty of Engineering and Technology

Name of the Student	Shikhar singh
Reg. No	17ETCS002168
Semester/Year	6th Semester / 3rd Year
Course Leader/s	Dr. Subarna Chatterjee

Declaration Sheet

Student Name	Shikhar singh		
Reg. No	17ETCS002168		
Programme	B.tech	Semester/Year	6th Semester / 3rd Year
Course Code	CSC309A		
Course Title	Computer Graphics		
Course Date		to	
Course Leader	Dr. Subarna Chatterjee		

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

Contents

Declaration Sheet ii

Contents iii

List of Figures 4

Question No. 1 5

1.1 Introduction: 5

1.2 Implementation of transformation: 6

1.3 Results with screenshots and discussion: 7

List of Figures

Figure 1: pentagon to be transformed..5
Figure 2:program function..6
Figure 3: main function...6
Figure 4: original polygon plot...7
Figure 5: scaled polygon..8
Figure 6: applying rotation 8
Figure 7: applying translation...9

Question No. 1

Solution to Question No. 1:

1.1 Introduction:

In computer Graphics, we have five types of transformation, i.e.

1. Translation – Moving an object from one point to another.
2. Rotation – Rotating an object.
3. Scaling – Changing the size of the object.

The given problem statement asks us to transform a polygon, i.e. to perform translation, scaling and rotation using opengl. Here, I have chosen a **pentagon** to perform transformation.

The coordinates of the chosen pentagon is:

1. -1.4, -2.0
2. 1.2, -2.0
3. 2.1, 0.5
4. 0.0, 2.0
5. -2.1, 0.5

Now this pentagon is to be rotated by **90º**, later for that rotated triangle we need to perform translation by 4 units in X direction and 8 units in Y direction. Finally, after performing rotation and translation, we need to scale the transformed pentagon by 5 units in x direction and 6 units in y direction. Later after performing all the three operations the final object will be displayed on the screen. Doing this is not that difficult as all prebuilt command are available for opengl that we need to use.

The chosen pentagon with above mentioned coordinates looks like this:

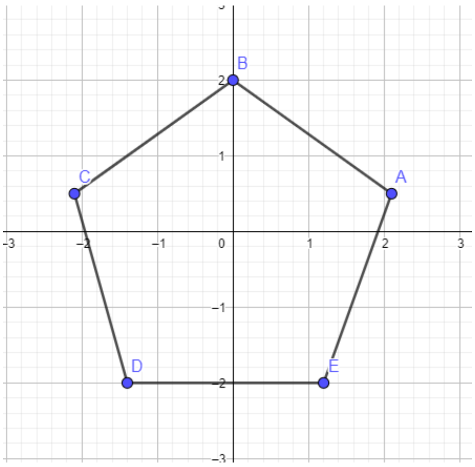
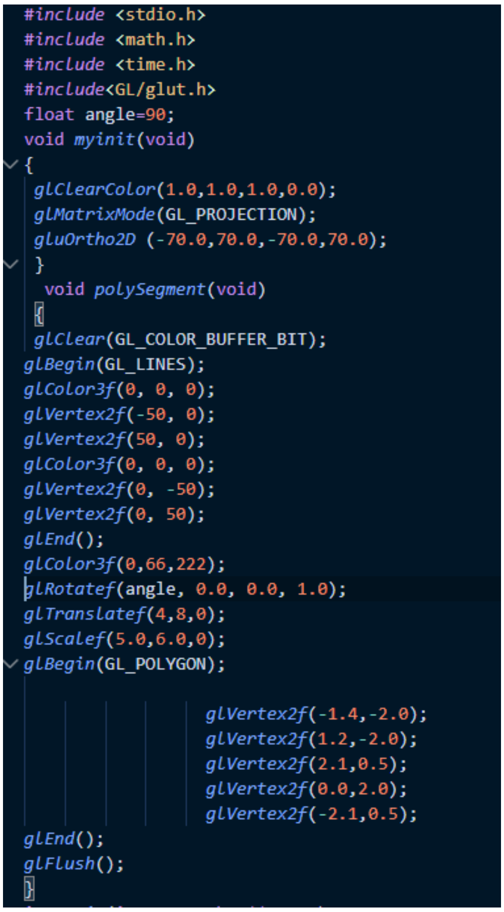


Figure 1: pentagon to be transformed

1.2 Implementation of transformation:

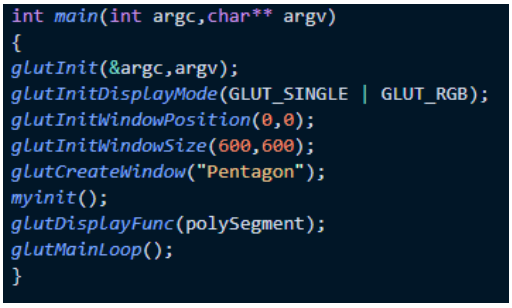
The code was implemented and executed in visual studio. The screenshot of the code is as follows:



```
#include <stdio.h>
#include <math.h>
#include <time.h>
#include<GL/glut.h>
float angle=90;
void myinit(void)
{
 glClearColor(1.0,1.0,1.0,0.0);
 glMatrixMode(GL_PROJECTION);
 gluOrtho2D (-70.0,70.0,-70.0,70.0);
 }
  void polySegment(void)
 {
 glClear(GL_COLOR_BUFFER_BIT);
glBegin(GL_LINES);
glColor3f(0, 0, 0);
glVertex2f(-50, 0);
glVertex2f(50, 0);
glColor3f(0, 0, 0);
glVertex2f(0, -50);
glVertex2f(0, 50);
glEnd();
glColor3f(0,66,222);
glRotatef(angle, 0.0, 0.0, 1.0);
glTranslatef(4,8,0);
glScalef(5.0,6.0,0);
glBegin(GL_POLYGON);

                  glVertex2f(-1.4,-2.0);
                  glVertex2f(1.2,-2.0);
                  glVertex2f(2.1,0.5);
                  glVertex2f(0.0,2.0);
                  glVertex2f(-2.1,0.5);
glEnd();
glFlush();
}
```

Figure 2:program function



```
int main(int argc,char** argv)
{
glutInit(&argc,argv);
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowPosition(0,0);
glutInitWindowSize(600,600);
glutCreateWindow("Pentagon");
myinit();
glutDisplayFunc(polySegment);
glutMainLoop();
}
```

Figure 3: main function

All the above code snippets show the transformation of the chosen polygon.

Function explanation: -

Void initGL(void) – This function is used to initialize opengl.

glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT) – This function clears the screen and depth buffers.

glClearColor() – This function sets background color to white.

glMatrixMode(GL_PROJECTION) – This function is used to select Projection matrix.

Int main() – This function makes the GLUT runs as a console application.

glutInit(&argc,argv) – This function is used to initialize GLUT.

glutInitDisplayMode() - This function is used to set display modes.

glutInitWindowSize()– This function is used to initialize window width and height.

glutCreateWindow() – This function creates a window with the given title.

glutDisplayFunc(display) – This function register handler for window repaint.

initGL() – This function initializes our own OpenGl.

glutMainLoop() – This function makes us enter the infinitely event handling loop.

The 3 important functions required as per this problem are:

glRotatef(angle) – used to rotate the polygon

glTranslatef(x,y,z) – to translate the polygon as per the given arguments i.e. x, y and z coordinates.

glScalef(x,y,z) – scale the polygon as per the given arguments i.e. x, y and z coordinates.

1.3 Results with screenshots and discussion:

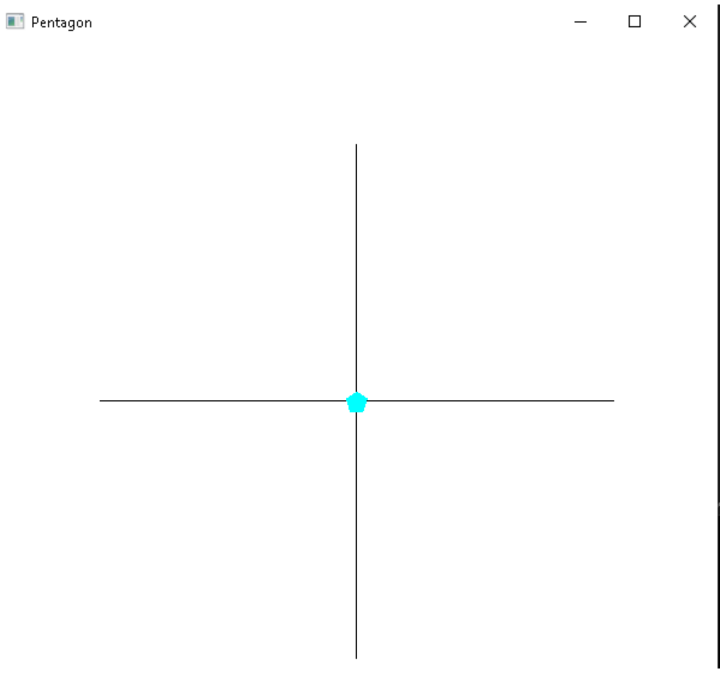


Figure 4: original polygon plot

The above fig-4 displays the original pentagon along the x-axis and y-axis as per the given co-ordinates.

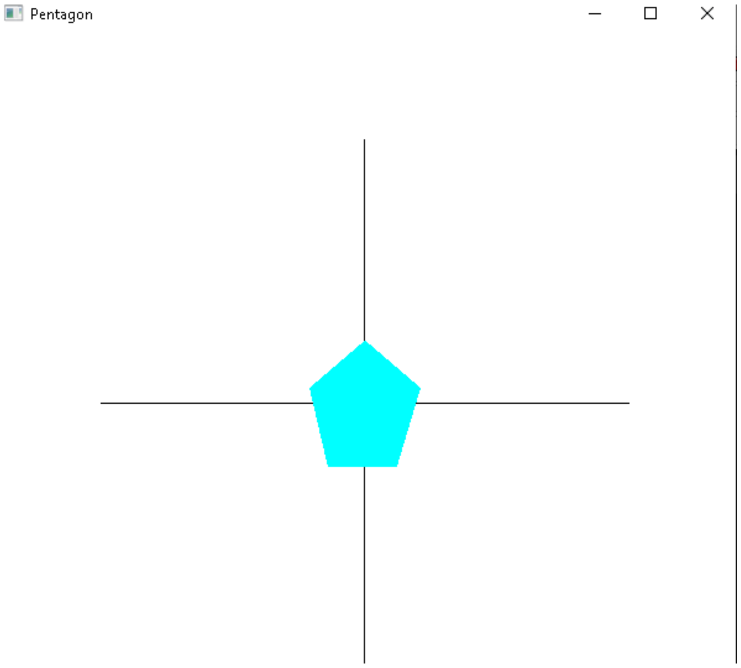


Figure 5: scaled polygon

Figure-5 shows the polygon after being scaled up as per the given requirements., i.e. 5 units in x-axis and 6 units in y-axis.

NOTE: - This figure is scaled first instead of applying rotation and translation first in order to see the figure clearly.

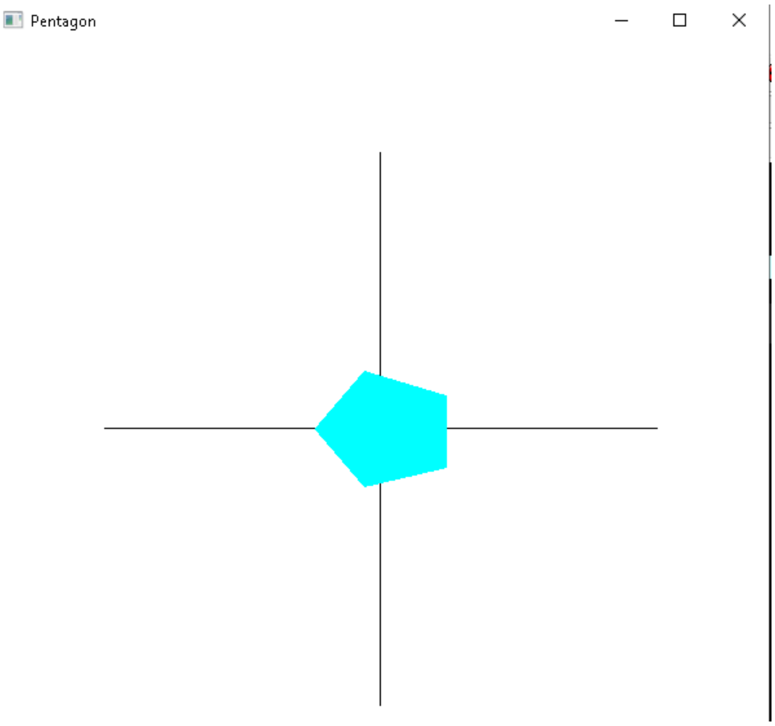


Figure 6: applying rotation

Figure 6 shows the transformed pentagon after applying rotation to it. We have rotated the original pentagon to 90 degree about the origin and then display it.

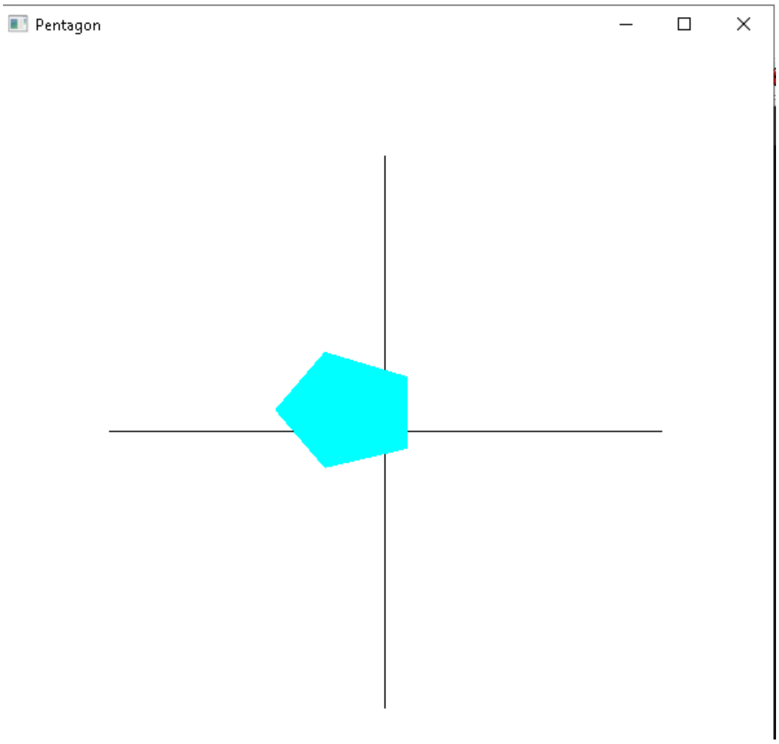


Figure 7: applying translation

The above figure – 7 shows the pentagon after applying translation about 4 units in x-axis and 8 units in the y-axis into the rotated and scaled pentagon. Translation means we have moved the object, i.e. pentagon from one position to other.

Also, this is the final result produced after applying rotation, translation and scaling to the original polygon chosen.