Question 1: (16 points) OpenGL/GLUT, Basic Definitions and Short Questions

a) (4 points) What is the framebuffer?

b) (4 points) What are Vertex Buffer Objects?

c) (4 points) What is a scene graph?

d) (4 points) Draw the shape generated by following OpenGL command

\texttt{glDrawArrays(GL\_TRIANGLE\_FAN, … )}
**Question 2: OpenGL/GLUT (10 points)**

a. (4 points) OpenGL has similarities with the C programming language. However, even though data types such as `int` and `double` are already defined in C, OpenGL redefines these data types. Why?

b. What are the GLUT commands to:

   i. (3 points) Register a keyboard callback function

   ii. (3 points) Register a reshape callback function

**Question 3: (10 points) Fractals/Mandelbrot Set**

The dwell function in the Mandelbrot set is the main function that is used to evaluate whether input complex numbers are in the Mandelbrot set or not. Two of your friends in CS 543 are arguing about the dwell function in the Mandelbrot set. Friend A says that he wanted a really accurate image, so he made his dwell function iterate 20000 times. Friend B says that she felt that 269 iterations were enough and that 20000 iterations were just a waste of time. These two friends come to you for an opinion. They want to know:

a. (5 points) Does the accuracy of the Mandelbrot set image increase infinitely as you increase the number of iterations? Specifically, is the Mandelbrot image more accurate for 20000 iterations than for 269 iterations?

b. (5 points) What number of iterations would you set your dwell function to and why?
Question 4: Transformations (26 points)

a. (14 points) Write out by hand the sequence of 4x4 matrices corresponding to the transformations commands listed below (in order). Multiply these matrices together to get the final Composite Transform Matrix (CTM). Show all your work.

\[ t = \text{Translate}(2,4,8); \]
\[ s = \text{Scale}(3,7,9); \]
\[ r = \text{RotateY}(40); \quad //\text{This is a Y-Roll by 40 degrees} \]
\[ \text{ctm} = t \times s \times r; \]

b. (4 points) What if the user added a pushMatrix() command right after the RotateY(40) command above, what effect would this have on the CTM?
a. (8 points) You would like your OpenGL program to declare a float variable `temperature` in your application and pass it to your vertex shader. In the vertex shader, this same variable is called `temp`. Give snippets of code snippets for declaring these variables in the application and vertex shader and connecting these two variables. Clearly indicate what snippets would go in the .cpp file and what snippets would go into the vertex shader.
**Question 5 (15 points): Linear Algebra for Computer Graphics**

Find the angle between the 2 vectors \( \mathbf{A} = (3,4) \) and \( \mathbf{B} = (4,3) \) using the dot product method.

**Question 6: Meshes (23 points)**

a) (5 points) What problem results when drawing meshes stored in a vertex list that is solved using an edge list?
b) (18 points) Using the Newell method for finding surface normals of a polygon, find the normal to a polygon with sides (2,6,0), (0,3,3), (4,2,1). Show all your work!