1. [20 points] Consider the following OpenGL commands:

```c
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glRotatef( 90.0, 0.0, 1.0, 0.0 );
glTranslatef( 2.0, 0.0, 0.0 );
glScalef( 2.0, 1.0, 1.0 );
```

What will the $4 \times 4$ modelview matrix be equal to after these commands have executed?

2. [20 points]

a. Describe the purpose of the specular exponent $f$. How is it used in the Phong lighting equation? (Give the relevant portion of the lighting equation.) What is the qualitative effect of making the specular exponent larger or smaller?

b. OpenGL requires that the specular exponent be the same for all wavelengths; that is, the same for red, green, and blue. Explain why this is. Hint: What would happen if the specular exponent for red was set to be significantly lower than the lower the specular exponent for green and blue?
3. [10 points] Give the definition of **linear transformation**.

4. [20 points] Let $f : \mathbb{R}^2 \to \mathbb{R}^2$ be defined to be the transformation that reflects points across the line $y = x - 1$. In particular, it maps a “F” as shown in the picture below.

Give the matrix that represents $f$ over homogeneous coordinates.
1. [20 points] Let \( f : \mathbb{R}^2 \to \mathbb{R}^2 \) be defined to be the affine transformation that maps an “F” as shown in the picture below.

\[
\begin{align*}
x & \quad y \\
\langle 0, -1 \rangle & \quad \langle 1, 1 \rangle \\
\langle 0, 0 \rangle & \quad \langle 2, 0 \rangle \\
\langle 2, 2 \rangle & \quad \langle 3, 1 \rangle
\end{align*}
\]

\( f \)

\( \implies \)

\( y \)

\( x \)

a. Is \( f \) a rigid transformation? Explain why or why not.

b. Express \( f \) in the form \( f(x) = Mx + b \) with \( M \) a \( 2 \times 2 \) matrix.

c. Give a sequence of “pseudo” OpenGL commands that will draw the “F” in the position shown on the right. Use commands such as \texttt{drawF()} (draws “F” in the position shown on the left), \texttt{pglRotatef(\ldots)}, \texttt{glTranslatef(\ldots)}, \texttt{glLoadIdentity()}, and \texttt{pglScale2f(\ldots)}. 
2. [15 points] Consider the following $3 \times 3$ matrix $M$ that operates on the homogeneous coordinates of points in $\mathbb{R}^2$.

\[
\begin{pmatrix}
-2 & -2 & 2 \\
4 & 0 & 0 \\
0 & 0 & 2
\end{pmatrix}.
\]

In the empty graph on the right, draw the image of the “F” under the affine map on $\mathbb{R}^2$ that is defined by the matrix $M$. Draw to scale, and label points as needed.

3. [10 points] Let $x = (0,0)$, $y = (1,0)$, $z = (2,1)$, and $w = (0,2)$. Under bilinear interpolation:

a. What point is obtained with $\alpha = 0$ and $\beta = 1$?

b. What point is obtained with $\alpha = 1$ and $\beta = 0$?

c. What point is obtained with $\alpha = \frac{1}{4}$ and $\beta = \frac{1}{2}$?

Give a sketch of the quadrangle defined by $x, y, z, w$ and the points which are the answers to questions a.-c.
1. [20 points] This problem concerns transformations in $\mathbb{R}^3$. Suppose you are given a function $\text{DrawCone()}$ that draws a cone of height 1, and base radius 1. This cone drawn by $\text{DrawCone()}$ is situated centered around the $y$-axis with its base on the $xz$ plane and the tip of the cone at $(0,1,0)$.

a. Give a code fragment that will draw the cone as shown in the figure: the cone is to be drawn upside down, and with height 2 and base radius 2. Its tip is now at the origin; it is still centered around the $y$-axis.

Your code fragment that draws the cone may use any of the following pseudo-OpenGL commands: $\text{glMatrixMode()}$, $\text{glLoadIdentity()}$, $\text{glRotatef()}$, $\text{glTranslatef()}$, $\text{glLoadMatrix()}$, $\text{glMultMatrix()}$, $\text{glScalef()}$, and $\text{DrawCone()}$.

b. Give a $4 \times 4$ homogeneous matrix that gives the same transformation as is used in your answer for part a.
1. [20 points] This problem concerns transformations in $\mathbb{R}^2$. Suppose you are given a function $\text{DrawCircle()}$ that draws a unit circle centered at the origin (radius equals one). Give a code fragment that will draw an ellipse as shown in the figure. The length of the ellipse is $\ell$ and the width is $w$. One endpoint of the ellipse is at $\langle x_0, y_0 \rangle$ in $\mathbb{R}^2$, namely, one of the endpoints of the axis along which the length $\ell$ is measured. The ellipsoid is tilted at an angle $\theta$ (measured in degrees).

Your code fragment that draws the ellipse may use any of the following pseudo-OpenGL commands: $\text{glMatrixMode()}$, $\text{glLoadIdentity()}$, $\text{pglRotatef()}$, $\text{pglTranslatef()}$, $\text{pglLoadMatrix()}$, $\text{pglMultMatrix()}$, $\text{pglScalef()}$, and $\text{DrawCircle()}$. 