Name: ______________________ **Student ID:** ______________________

1) In terms of a sequence of transforms (translate, rotateX, rotateY, rotateZ), give the view transform from world space to camera space corresponding to a camera at world space point $(2, 5, 0)$ and looking straight at world space point $(10, 5, 0)$.

$$x_{\text{cam}} = \text{RotateY}(90^\circ) \cdot \text{Translate}(-2, -5, 0) \cdot x_{\text{ws}}$$

2) What are homogeneous (4D) coordinates, and how can you convert back and forth with regular 3D coordinates for a point?

Homogeneous coordinates are an extension of regular coordinates with the addition of an extra dimension, allowing more convenient transforms.

To go from regular 3D coordinates (x, y, z) for a point, add a fourth coordinate 1:

$$(x, y, z) \to (x, y, z, 1)$$

To go from homogeneous coordinates back to regular 3D, divide by the fourth coordinate:

$$(x, y, z, w) \to (x/w,\ y/w,\ z/w)$$

3) Why do homogeneous coordinates make translation transforms more convenient? Make sure to include a matrix in your explanation.

They allow translation (addition of a fixed vector) to be expressed with matrix multiplication, allowing easy composition/inversion of translation with other affine transforms. The translation of (x, y, z) by vector (a, b, c), i.e. to point $(x + a, y + b, z + c)$ can be expressed as:

$$\begin{pmatrix} x+a \\ y+b \\ z+c \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

4) Why do homogeneous coordinates make perspective transforms more convenient? Make sure to include a matrix in your explanation.

The conversion from homogeneous vectors back to regular 3D coordinates introduces a natural division; division by camera z is the crucial feature of perspective transformation, so we can bring it into the matrix composition framework:

$$\begin{pmatrix} -x/z \\ -y/z \\ -(z+1)/z \end{pmatrix} \leftarrow \begin{pmatrix} x \\ y \\ z+1 \\ -z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

5) Sketch an example of a triangle where our rasterization algorithm is extremely inefficient.

I'm going to skip trying to draw this diagram. The best example is a very skinny triangle which doesn't contain any pixel centres, but is almost aligned with a diagonal through the entire image. The bounding box of this triangle is the entire image, but no pixels end up being set.

Name: ______________________ **Student ID:** ______________________

6) Give pseudocode for the Z-Buffer algorithm.

- Set every Z-buffer entry $Z(i, j)$ to maximum depth value.
- For every triangle T:
- For every pixel $P = (i, j)$ in the bounding box of T:
- If P is inside T:
- Interpolate the depth z of the fragment from triangle corners.
- If $z < Z(i, j)$:
- Set $Z(i, j) \leftarrow z$.
- Shade the fragment and set colour of P to the fragment.

7) Give pseudocode (with formulas) for testing if 2D point (x, y) is inside a triangle with corners (x_0, y_0), (x_1, y_1), and (x_2, y_2).

First evaluate the three edge functions:

$$\begin{aligned} f_{01} &= (y - y_0)(x_1 - x_0) - (x - x_0)(y_1 - y_0) \\ f_{12} &= (y - y_1)(x_2 - x_1) - (x - x_1)(y_2 - y_1) \\ f_{20} &= (y - y_2)(x_0 - x_2) - (x - x_2)(y_0 - y_2) \end{aligned}$$

- If $f_{01} \geq 0$ and $f_{12} \geq 0$ and $f_{20} \geq 0$:
- Return true.
- Else if $f_{01} \leq 0$ and $f_{12} \leq 0$ and $f_{20} \leq 0$:
- Return true.
- Else:
- Return false.

8) **[Challenge]** Consider the barycentric coordinate α of a point (x, y) in a triangle. In what direction does the gradient vector $\nabla\alpha = (\partial\alpha/\partial x\ , \partial\alpha/\partial y)$ point? Illustrate with a sketch.

No sketch in this text file, but...

The barycentric coordinate α varies linearly, thus has a constant gradient. Furthermore, α is zero all along the edge opposite to the vertex to which it's associated, i.e. the opposite edge is an isocontour. Therefore the gradient $\nabla\alpha$ is orthogonal to the opposite edge. Finally, α increases from zero on the opposite edge to one at the vertex, so the gradient must be pointing towards the vertex (and perpendicular to the opposite edge).