1. (4) What effects and advantages does ray-tracing provide over projection based graphics? Why is ray-tracing not used in commodity graphics? Just write two sentences.

*Photorealistic images*

*Too expensive*

2. (4) Why does a simple pinhole camera work? A sentence please.

*It emulates perspective transform*

3. (4) Find the projection \( x_p, y_p, z_p \) for the measurement \( d=1 \) (along -z axis)

\[(x/z, y/z, -1)\]

4. (4) Describe in a sentence each what happens when you

(a) increase resolution of a raster image

*MORE NUMBER OF PIXELS IN IMAGE AND HENCE BETTER SAMPLING*

(b) increase the depth of pixels

*MORE RANGE OF COLORS AND BETTER CLARITY*

5. (4) Sketch the rendering pipeline with all stages. For each stage what is the input and output. Indicate on the sketch.

6. (4) Where is the bottleneck in the rendering pipeline if the scene is mostly composed of geometry? And where is the bottleneck if the scene is texture rich.

*Vertex shader*

*Fragment shader*
7. (4) List two reasons why GPUs are superfast?
   Multi-cores
   Streaming architecture

8. (4) What mode of drawing is encouraged in GLSL? Provide a brief explanation.
   Retained mode
   Asynchronous mode helps with execution

9. (4) What should be the GL_token that needs to be provided to gl.DrawArrays to render the following bunnies. Look closely the primitives. The left is a bunch of small surface polygons and the rest is a cloud. Go back to the Sierpinski Gasket examples.

   ![Bunnies](image)
   
   GL_TRIANGLES   GL_POINTS

10. (4) Sketch a typical GLSL program. For inspiration think of init() for Sierpinski gaskets.

   ```
   Main Web GL program {
     // compute and store new primitives
     // Create a vertex array object
     // Create and initialize a buffer object
     // Load shaders and use the resulting shader program
     // Initialize the vertex position attribute from the vertex shader
   }
   
   Vertex and fragment shaders pass in and out variables and compute with them```