A1. Computer Graphics (25 points)

Texture Mapping:

1. [10 points] Consider a cylinder aligned with the Y axis, running from y = 0 to y = 5, with radius 1. You have a "stone" texture map whose s and t coordinates are in the range [0, 1] and which you want to wrap around the cylinder to create a "pillar" model. Give the inverse mapping function f which maps a point (x, y, z) on the cylinder to the corresponding point (s, t) on the texture. Show your work.

2. [3 points] Now suppose you want to create a realistic (yet static) lighting effect on your stone pillar using radiosity or some other global method that is too expensive to render for a real-time game application. How could you use textures to solve this problem?

3. [10 points] What are MIPMAPS? Explain in detail when and how they can be used to reduce texture aliasing.

4. [2 points] If a texture map is stored as a W x H image in 24-bit color, how much additional storage is necessary to make a MIPMAP from it?

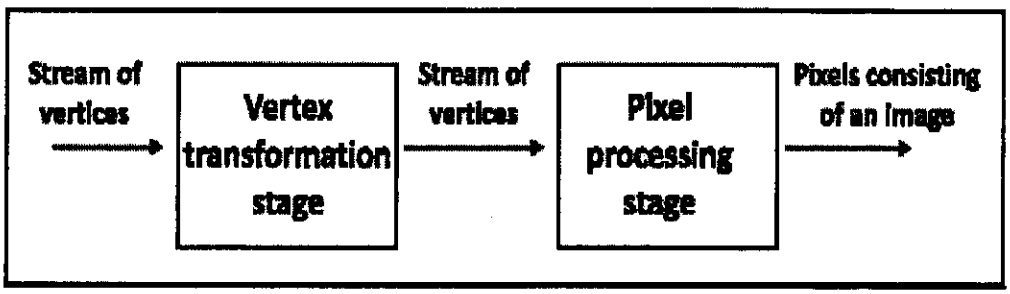
A2. Computer Graphics (25 points)

Ray Tracing: Consider a ray r that originates from the origin (0, 0, 0) and has direction (0, 0, 1) (along z-axis). Given a sphere centered at (x0, y0, z0) with radius 1.

1. [5 points]. Derive when the ray r will intersect the sphere.
2. [6 points]. Derive the intersection point if r intersects the sphere.
3. [9 points]. Assume the sphere is reflective, derive the reflected ray direction.
4. [5 points]. A typical ray tracing implementation usually offsets the origin of the reflected ray slightly off the intersection point along the reflection direction. Why is that?

A3. Computer Graphics (25 points)

Rendering Pipeline:



The traditional rendering pipeline has many stages. We can simplify those stages into two following two major steps: 1) vertex transformation stage (e.g., modeling and viewing transformation) step, and 2) pixel processing stage (e.g., rasterization). Since these two stages can be performed independently, we can perform these two stages in a parallel manner in graphics hardware.

1. [5 points]. If we move view-frustum culling right before the vertex transformation stage, will it improve the performance? Explain why or why not.

2. [6 points]. Assume that we perform Gouroud shading. If we increase the image resolution from 500x500 to 1Kx1K, in practice we will not observe any performance degradation in terms of total rendering time. Explain why.

3. [7 points]. If we perform Phong shading via surface subdivision, will we observe perform degradation if we double the image resolution? And if so, which stage(s) cause the degradation and why?

4. [7 points]. If we perform Phong shading using the shaders, will we observe perform degradation if we double the image resolution? And if so, which stage(s) cause the degradation and why?

A4. Computer Graphics (25 points)

Shading and Transformations:

1. [4 points] Consider a 24 bit per pixel full color RGB raster graphics system having 2048 by 2048 frame buffer. Assuming a rate of 290 megabytes per second, what would be the maximum achievable frame rate?

2. [2 points] State the reason for having two color buffers, one to rasterize while the other gets copied onto the screen.

3. [4 points] Explain what is a texture object in OpenGL and its benefit?

4. [6 points] Give the Phong's Reflection Model and Explain clearly all the terms involved.

5. [9 points] (i) What is a Rigid Body Transformation? (ii) Give two examples of Non-Rigid Transforms by way of Homogenous matrices, and state what they do. (iii) Rotate the point (3, 2, 1) by 45 degrees about the axis (9, 6, 3).