You have 70 minutes. No computers, notes, or other aids are allowed.

The best answers are not just factually and logically correct. They include all important information, while excluding irrelevant information. They convince the grader that the student understands the context and motivation. They are concise. Frequently they employ diagrams or pictures.

Good luck.

A. Draw a smiley face at the origin of a coordinate system. Give it a tattoo or earring, so that it is not symmetric. Write down a specific 2D translation vector $\langle t_0, t_1 \rangle$ and rotation angle θ. Draw the smiley face rotated-then-translated, and draw it again translated-then-rotated, to illustrate the distinction.

B. Suppose that you have a 3D scene graph, that contains a mesh, that contains a triangle with attributes $\vec{a}$, $\vec{b}$, $\vec{c}$. In our finished 3D graphics engine, a multi-step process transforms these vectors into the vectors that are passed to `triRender`. List the steps in order. For each step, describe its purpose or meaning in a single sentence, using terminology as appropriate.

C. Now that we have a 3D graphics engine with projections, clipping, and everything else, do we still have a 2D graphics engine? Could you make a 2D game in main.c, using the most recent versions of our other files, without editing those other files? Explain.

D. In a scene graph, why do we use 3×3 (for 2D) or 4×4 (for 3D) matrices? Is it possible to implement a scene graph system without them?

E. These questions concern the triangle-level clipping algorithm that we use at the near plane — not other clipping algorithms, such as pixel-level clipping in the depth buffer.

E0. Why must we clip at the near plane?

E1. Why might we want to clip at the other planes using the same algorithm?

E2. If we were to clip at all planes bounding the viewing volume, then how many triangles might the clipping system emit, for each triangle that enters it?