1. *(2 pts)* (Reverse engineering rotations) In 2D, if the point (3,2) rotates about the origin to (2,-3), what's the angle?
2. *(2 pts)* Give the matrix M that has this property: for all vectors p, *Mp*=⎛⎝⎜⎜345⎞⎠⎟⎟×*p*Mp=(345)×p.
3. *(2 pts)* Give the matrix M that has this property: for all vectors p, *Mp*=⎛⎝⎜⎜345⎞⎠⎟⎟⋅*p*Mp=(345)⋅p.
4. *(2 pts)* Why can the following not possibly be a 3D Cartesian rotation matrix?

⎛⎝⎜⎜210000001⎞⎠⎟⎟(200100001)

1. *(2 pts)* Use any method (not involving soliciting answers on the internet) to rotate the point (4,4,6) by 60 degrees about the axis (2,2,3). Explain your method. (E.g., if you saw the answer in a vision, are your visions generally accurate?)
2. *(2 pts)* Can the volume of a small cube change when its vertices are rotated? (yes or no). Why (not)?

*No, volume is the product of the sides of the cube multiplied by the cubes determinant. Since the determinant and the sides do not chanfed when the vertices rotate it cannot change the volume. Furthermore objects generally don’t change in volume from rotating so common sense dictates this.*

1. *(2 pts)* What is the ''event loop''?

An infinite loop entered by your program. It waits for events such as state change sor user inputs.For any event the corresponding event handler is called from the loop which will eventually return to the event loop.

1. *(2 pts)* Why does putting all your vertices into an array and telling OpenGL about it make a big graphics program faster?

This way you don’t need to keep doing subroutine calls which saves memory. The vertices may be cached in the GPU which would reduce data on the graphicsbus.

1. *(2 pts)* Since the Z (aka depth) buffer looks so useful, why is it not enabled by default?

There are situations where it may be faster not to use the depth buffer for example if you have a flat image.

1. *(2 pts)* What's the quaternion representing a rotation of 180 degrees about the axis (0,0,1)?
2. *(2 pts)* Use the quaternion formulation to rotate the point (0,1,0) by 180 degrees about the axis (0,0,1).
3. *(2 pts)* Use the vector formulation to rotate the point (0,1,0) by 180 degrees about the axis (0,0,1).

(0,-1,0)

1. *(26)* Extend your program from last week that displays the Starship Enterprise as follows:
   1. Do the rotation in the vertex shader instead of in the javascript program.
   2. Make the color of each pixel depend on its z-value.