1. Answer the questions below by specifying the appropriate option i.-v. The questions assume the shader program has only a vertex shader and fragment shader, and also that the shaders do not access any buffers or textures.

a. When Gouraud interpolation is used, the Phong lighting calculation is done:  
   (Answer i., ii., iii., iv., or v.)
   i. Before the vertex shader is called.
   ii. By the vertex shader.
   iii. After the vertex shader is called but before the fragment shader is called.
   iv. By the fragment shader.
   v. After the fragment shader is called.

b. When Phong interpolation is used, the Phong lighting calculation is done:
   i. Before the vertex shader is called.
   ii. By the vertex shader.
   iii. After the vertex shader is called but before the fragment shader is called.
   iv. By the fragment shader.
   v. After the fragment shader is called.

c. When Gouraud interpolation is used, the Gouraud interpolation calculation is done:
   i. Before the vertex shader is called.
   ii. By the vertex shader.
   iii. After the vertex shader is called but before the fragment shader is called.
   iv. By the fragment shader.
   v. After the fragment shader is called.

d. For parts d., e.f., two vertices are considered to be the “same vertex” if they have exactly the same values for their vertex attributes (even if they have different vertex ID’s). Recall that `glDrawElements` uses an element array (an EBO), but `glDrawArrays` does not.

   Suppose `glDrawArrays` is used to render vertices from data in a VBO using `GL_TRIANGLES`. If the same vertex \( v \) appears as a vertex in multiple triangles, then a vertex shader is invoked for \( v \):
   i. At most once, possibly not at all.
ii. Exactly once.
iii. One or more times, but possibly not for every triangle containing v.
iv. Exactly once per triangle containing v.

e. Now suppose `glDrawArrays` is used to render vertices from data in a VBO using `GL_TRIANGLES_STRIP`. If the same vertex \( v \) appears as a vertex in multiple triangles, then a vertex shader is invoked for \( v \):

i. At most once, possibly not at all.
ii. Exactly once.
iii. One or more times, but possibly not for every triangle containing \( v \).
iv. Exactly once per triangle containing \( v \).

f. Suppose `glDrawElements` is used to render vertices from data in a VBO and EBO using `GL_TRIANGLES_STRIP`. If the same vertex \( v \) appears as a vertex in multiple triangles, then a vertex shader is invoked for \( v \):

i. At most once, possibly not at all.
ii. Exactly once.
iii. One or more times, but possibly not for every triangle containing \( v \).
iv. Exactly once per triangle containing \( v \).

2. For parts d., e., f. of the previous question, two vertices were considered to be the “same vertex” if they have exactly the same values for their vertex attributes (even if they have different vertex ID’s). Suppose we changed the convention, and instead consider two vertices to be the “same vertex” only if they have the same vertex ID? Do parts d., e., f. of Problem 1 still make sense under this revised definition “same vertex”? How would you answers to these three questions change under this revised definition?