Please read these instructions carefully!

- Answer the test questions in the separate answer sheet.
- You may use the back of all the paper sheets as drafting area (*rascunho*).
- How to:

	A	B	C	D	E
Select the answer (A):	●	○	○	○	○
Replace the answer (A) by (C):	✗	○	●	○	○
Cancel (C) and reactivate (A):	⊗	○	✗	○	○

- This test has 33 QUESTIONS (ignoring question zero), each question has a score of 20/33 points.
- **POINTS LOST** for each wrong answer (in percentage of the question's value):
  $1st = 0\%$ $2nd = 11.11\%$ $3rd = 22.22\%$ $4th+ = 33.33\%$.

Name: ______________________________ Number: ____________

0. Test version. This is not a question and will be ignored for grading!

   A. ⇒ ● ⇐ B. C. D.

1. Consider the pipeline of a modern graphics card, exposed to a 3D WebGL program. Choose the most appropriate description for the task that gets executed in the first programmable stage of the pipeline.

   A. Decide if the corresponding pixel is going to be painted or not and, in this later case, assign it with its final color.

   B. Determine the final location of a vertex on the screen.

   C. Assign a color to the corresponding pixel.

   D. Determine the final location of a vertex in Clip Coordinates.

2. Consider the pipeline of a modern graphics card, exposed to a 3D WebGL program. Choose the correct stage of the pipeline where the operation known as '`rasterisation`' is executed.

   A. Immediately before clipping the primitive against the clip volume.

   B. Immediately after the creation of the primitive and its clipping against the clip volume.

   C. Immediately after the processing of each vertex.

   D. At the end of the pipeline, in its final stage.

3. Consider the pipeline of a modern graphics card, exposed to a 3D WebGL program. Choose the information that best describes what is associated with a fragment at the exit of the '`fragment processor`´.

   A. An `RGB()` or an `RGBA()` tuple, a location on the frame buffer and a depth value for z-buffering.

   B. An `RGB()` or an `RGBA()` tuple and a depth value for z-buffering.

   C. An `RGB()` or an `RGBA()` tuple.

   D. An `RGB()` or an `RGBA()` tuple and a location on the frame buffer.

4. Choose the option that describes the correct behaviour for a GLSL variable declared with the '`varying`' qualifier.

   A. Its value is computed in the fragment shader and it is interpolated for every vertex during the execution of the vertex shader

   B. Its value is computed inside the vertex shader and that same value is later accessible from the fragment shader.

   C. Its value is computed inside the vertex shader and it is later linearly interpolated during the rasterisation stage but it is not accessible from the fragment shader.

D. Its value is computed inside the vertex shader and it is later linearly interpolated during the rasterisation stage for each fragment. This interpolated value is then accessible during the fragment processing phase.

5. Choose the option that describes the correct semantics for the value of a GLSL variable declared in a fragment shader with the ‘`attribute`’ qualifier.

   A. None of the other alternatives are correct.

   B. Its value varies from fragment to fragment, by linear interpolation of the computed values for each vertex of the primitive being rendered.

   C. Its value varies from vertex to vertex and it is extracted from a buffer filled by the javascript side.

   D. Its value is assigned a single common value in the javascript side and it remains fixed during the GLSL program execution.

6. Consider the GLSL piece of code presented on the right. Local variables ‘`c`’ and ‘`p`’ have previously been declared with the correct type to make the presented code valid in GLSL. What is the value that gets assigned to ‘`p`’?

```
c = 2.0 * vec3(0.5, 2.0, 1.0);
p = dot(c.zzx, vec3(4.0, c.b, 6.0));
```

   A. (4.0, 2.0, 6.0) B. (8.0, 4.0, 6.0) C. 18.0 D. 12

7. Consider the following pair of shaders, written in GLSL and presented on the right. They are used by an application that draws a model of a car engine part using both color and transparency. The color and transparency values are assigned by the javascript application on a per vertex basis. Which of the following sequences of words can be used to fill the blanks in the code so that the application works as described?

```
// vertex shader
attribute vec3 v;
_________ vec4 w;
attribute float f;
_________ vec3 g;

main()
{
        gl_Position = v;
        ____ = vec4(____, ____);
}

// fragment shader
varying vec4 w;
main()
{
        gl_FragColor = w;
}
```

   A. `varying, attribute, w, g, f`

   B. `varying, attribute, w, f, g`

   C. `uniform, varying, w, g, f`

   D. `uniform, varying, w, f, g`

8. A WebGL application needs to implement a melting effect. The objective is to display an animation of an object made up of triangular faces that will gradually collapse to the ground (y=0) over time, following a straight vertical path, without ever loosing their connectivity. Suppose that the buffers containing the geometry and the topology of the mesh have already been created. Where would you implement this melting operation to be performed as efficiently as possible?

   A. In the javascript part of the application, by modifying the contents of the buffer containing the topology information.

   B. Both in javascript part of the application, by modifying the buffer containing the geometry, and in the vertex shader, by applying an adequate transformation to the vertex.

   C. In the vertex buffer, by applying the adequate transformation to the vertex position based on an external uniform input variable.

   D. In the javascript part of the application, by modifying the contents of the buffer containing the gemetry.

9. A WebGL application needs to offer the user the possibility to choose one object from the set of visible objects on screen at a certain time. These objects, made up of several triangles, are being illuminated and shaded using the Phong illumination model as well as Phong shading, and we cannot allocate any additional WebGL data buffers to execute the

task. Choose the correct option below assuming that the picking operation will be implemented using an offscreen color buffer.

A. The method would work for both the visible objects and those that are occluded by them.

B. Each triangle of each object would be painted with a unique color.

C. The method would not work when rendering the illuminated scene because several pixels from different objects may end up having the same final colour in the frame buffer.

D. The number of drawing commands (or drawing calls) to WebGL would increase when compared to those required to draw the illuminated scene.

10. Consider the entities in the right table represented in homogeneous coordinates. What is the result of the expression: B + C − A + D?

A. (1, 5, 5, 1) B. (1, 7.5, 5, 6). C. (0, 7, 2, 5) D. (0, 7/5, 2/5, 1)

A	(2, 1, 6, 2)
B	(0, 3, 4, 0)
C	(2, 2, 4, 1)
D	(0, 3, 0, 6)

11. Consider the representation of points and vectors using homogeneous coordinates. Which option is correct?

A. Both points and vectors have a unique representation.

B. Both points and vectors have several distinct representations.

C. Each point has several representations but each vector has a unique representation.

D. Each point has a unique representation but each vector has several representations.

12. Choose the option that includes all the elementary 3D geometrical transformations that are considered as rigid body transformations (no changes in shape).

A. uniform shearing, uniform scaling and rotation.

B. translation and uniform scaling.

C. translation and rotation.

D. uniform scaling and rotation.

13. Which of the following composition of transformations can be considered the inverse transformation of:

$$T(3, 5, -4).Ry(-30).S(2, 4, 2)$$

A. $S(-2, -4, -2).R_y(30).T(3, 5, 4)$

B. $T(-3, -5, 4).R_y(30).S(1/2, 1/4, 1/2)$

C. none of others

D. $R_y(30).S(1/2, 1/4, 1/2).T(-3, -5, 4)$

14. The Modelview matrix is commonly used in the implementation of 3D graphics systems. Choose the option that correctly describes it.

A. It transforms points and vectors from World coordinates to Camera coordinates.

B. It transforms points and vectors from Object coordinates to World coordinates.

C. It transforms points from Object coordinates to Camera coordinates.

D. It transforms points from World coordinates to Camera coordinates.

15. A computer screen with FullHD resolution (1920 x 1080 pixels [16:9]) is going to be used to display the contents of a window defined in 2D world coordinates by its limits $-100 \leq x \leq 60$ and $400 \leq y \leq 480$. The viewport is supposed to aligned with the top right corner of the screen and it should maximize the viewing area. What are its dimensions in pixels?

    A. $(\frac{1080}{9} \times 16) \times 1080$ B. $1920 \times (\frac{1920}{16} \times 8)$. C. $(\frac{1080}{8} \times 16) \times 1080$. D. $1920 \times (\frac{1920}{16} \times 9)$.

16. With the conditions defined in Question 15, what would you choose as the first operation to be performed by the window to viewport transformation?

    A. $T(-60, -400)$ B. $T(-60, -480)$ C. $T(-1920, -1080)$ D. $T(-1920, 0)$

17. With the conditions defined in Question 15, what would you choose as the last operation to be performed by the window to viewport transformation?

    A. $T(60, 480)$ B. $T(1920, 0)$ C. $T(60, 400)$ D. $T(1920, 1080)$

18. Once again, with the conditions defined in Question 15, what would you change in the problem specification to visualize the same content in the same location of the screen but using its full 1920 x 1080 viewing area?

    A. The scaling factor included in the window to viewport transformation.
    B. The lower limit (ymin) of the window.
    C. The final translation in the window to viewport transformation.
    D. The upper limit (ymax) of the window.

19. Consider an implementation of the z-buffer algorithm that normalizes z values between the *front* and the *back* clipping planes to the range [0,1], respectively. Which of the following options is **correct** for a typical use of the algorithm for normal HLHSR.

    A. The buffer is cleared with a 1 value and a pixel passes the test if its z-value is lower than the stored value in the z-buffer.
    B. The buffer is cleared with a 1 value and a pixel passes the test if its z-value is greater than the stored value in the z-buffer.
    C. The buffer is cleared with a 0 value and a pixel passes the test if its z-value is greater than the stored value in the z-buffer.
    D. The buffer is cleared with a 0 value and a pixel passes the test if its z-value is lower than the stored value in the z-buffer.

20. The back face culling algorithm is enough to solve the problem of HLHSR when applied to certain scenes. Choose the option where this algorithm alone would **fail** to solve the HLHSR problem.

    A. A cube with a hole that was created by subtracting a cylinder with a CSG (constructive solid geometry operation).
    B. Several solid cubes that don't overlap in the projection.
    C. A single cylinder solid.
    D. A cube and a cone that overlap in the projection but not in 3D space and that are drawn in back to front order.

21. Which of the following expressions builds an axonometric projection?

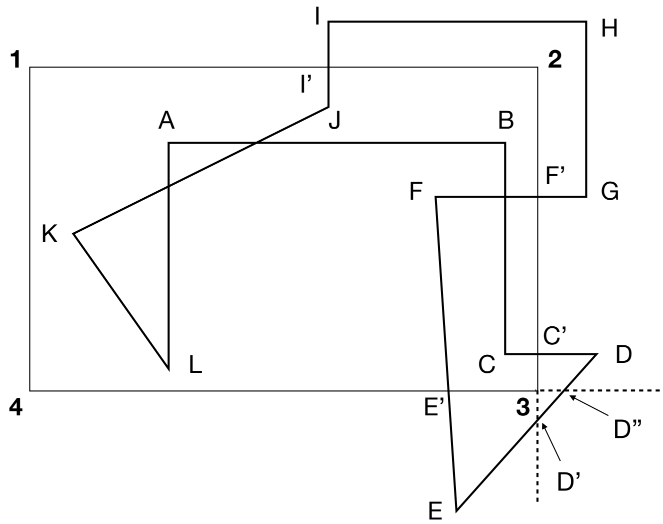
    A. $M_{ORT}.R_x(20).R_y(-30)$
    B. $M_{PER}.R_x(15).R_z(5)$

C. $M_{PER}.R_x(15).R_y(20)$
D. $M_{ORT}.R_x(10).R_z(-20)$

22. Which of the options in Question 21 would allow us to create a projection of an axis aligned cube with 3 vanishing points?

A. (C) B. (A) and (D) C. (A) and (B) D. (B) and (C)

23. Consider the polygon P=[A, B, C, D, E, F, G, H, I, J, K, L] represented on the right. The polygon P is going to be clipped with the Sutherland-Hodgeman polygon clipping algorithm against the window Q=[1, 2, 3, 4], using the following clipping order: Right, Bottom, Top, Left. What are the first 6 vertices of the output after the first clipping stage?



A. [B, C, C', D', E, F, ...]
B. [A, B, C, D, D", E', ...]
C. [B, C, D, D", E', F, ...]
D. [A, B, C, C', D', E, ...]

24. In the situation described in Question 23, what are the first 6 vertices after the second stage?

A. [C, C', D', E', F, F', ...]
B. [C, C', 3, E', F, F', ...]
C. [A, B, C, C', 3, E', ...]
D. [B, C, C', 3, E', F, ...]

25. How many edges would the final clipped polygon P' contain after all the clipping stages?

A. 15 B. 13 C. 14 D. 12

26. Now let us consider that the edges of that same polygon P, are going to be clipped by the Cohen-Sutherland line clipping algorithm. How many edges would be trivially accepted?

A. 5 B. 4 C. 3 D. 2

27. And how many edges would be trivially rejected under the conditions of Question 26?

A. 2 B. 3 C. none D. 1

28. Consider that polygon P is going to be painted with the Fill Area (or scanline) algorithm. How many non empty entries would the Edges Table contain?

A. 5 B. 7 C. 6 D. 4

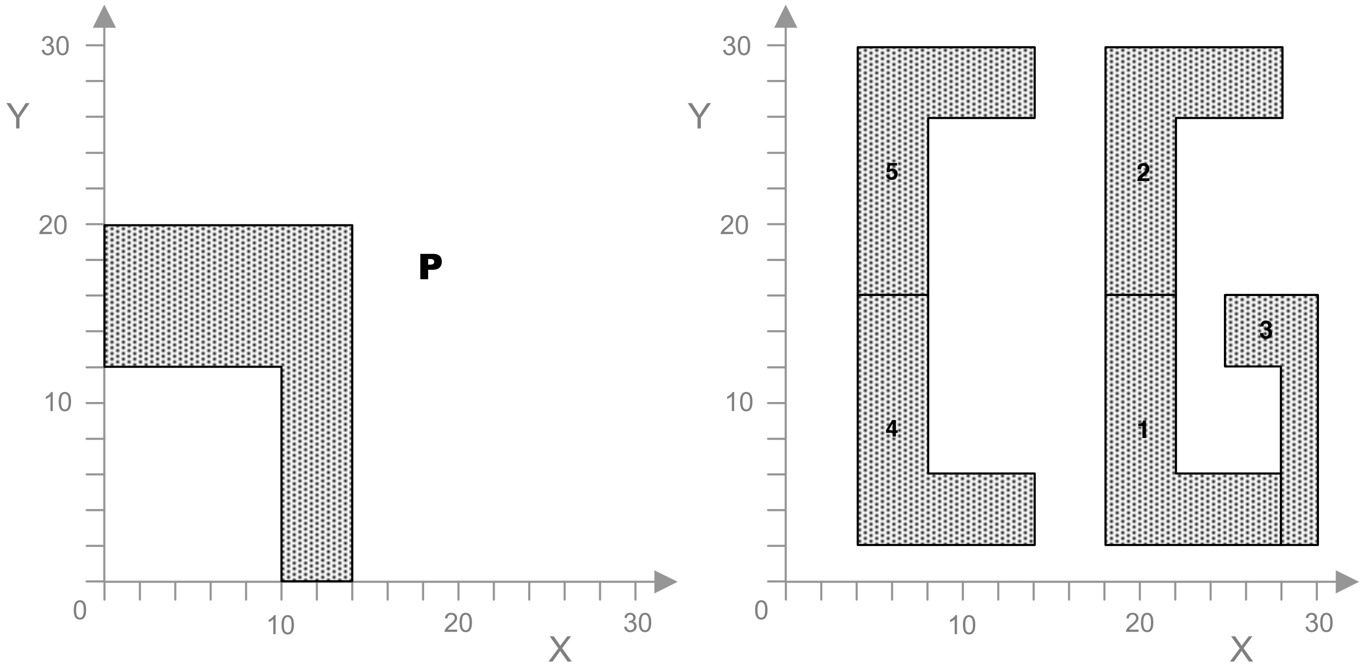
29. With respect to the scenario of Question 30, what is the total number of edges contained in the Edges Table?

A. 7 B. 8 C. 11 D. 12

30. With respect to the scenario of Question 30, what would be the content of the Active Edges Table immediately before painting the scanline that passes through vertex F?

A. $\overline{AL} \rightarrow \overline{KJ} \rightarrow \overline{EF} \rightarrow \overline{BC}$
B. $\overline{KJ} \rightarrow \overline{AL} \rightarrow \overline{EF} \rightarrow \overline{BC} \rightarrow \overline{GH}$
C. $\overline{KJ} \rightarrow \overline{AL} \rightarrow \overline{BC} \rightarrow \overline{GH}$
D. $\overline{AL} \rightarrow \overline{KJ} \rightarrow \overline{BC} \rightarrow \overline{GH}$

Take a look at the following figure. The 2D composition on the right side is based on the use of a single primitive **P**, shown on the left side of the figure. Consider only the problem of modelling the letter 'G', exactly as it appears in the figure.



31. What is the sequence of transformations to make part 1 of letter 'G'?

32. What is the sequence of transformations to make part 4 of letter 'C'?

33. What is the sequence of transformations to make part 5 of letter 'C'?