

I. (25p) Indicați răspunsul corect.

1. (1p) Un cod RGB care generează o nuanță de gri (**alegeți deschis sau închis**) este
2. (1p) O funcție GLSL utilizată în shading este
3. (1p) O funcție care are ca efect schimbarea sensului de parcurgere a unui poligon este
4. (1p) O funcție legată de folosirea indexării vârfurilor este
5. (1p) Pentru a putea decupa un trunchi de piramidă arbitrar se folosește funcția
6. (2p) La apelarea funcției `glDrawArrays(GL_TRIANGLES)` pentru un *Vertex Array Object* cu (**alegeți un număr $n \in \mathbb{N}^*$, $n \geq 30$**) vârfuri distincte, numărul de triunghiuri desenate este de
7. (2p) Dacă la reprezentarea unui segment folosind algoritmul Bresenham s-a obținut parametrul de decizie $p_0 = 6$, iar valoarea lui p_1 este (**alegeți valoarea lui p_1**), atunci panta m a dreptei suport este
8. (2p) Date punctele $M = (0, 0, 5)$, $N = (5, 0, 0)$, $P = (\alpha, \beta, \gamma)$ (**alegeți valorile, astfel ca P să nu fie situat pe dreapta MN**), un punct situat în interiorul triunghiului MNP este
9. (2p) Folosind regula indexului pentru o linie poligonală închisă, un punct interior este dat de valorile $n_+ =$, $n_- =$ (**alegeți valori**).
10. (2p) În planul proiectiv $\mathbb{P}^2\mathbb{R}$ punctul $[5 : -2 : \alpha]$ (**alegeți $\alpha \neq 0$**) coincide cu punctul (**alegeți, așa încât prima coordonată să fie ≥ 6**).
11. (2p) Determinantul matricei 4×4 asociate scalării de factor s (**alegeți $s \in \mathbb{R}^3$ cu toate componentele nenule**) este
12. (2p) O matrice 4×4 cu determinant nenul care permite schimbarea poziției dreptei de la infinit este
13. (2p) Un exemplu de puncte P_0 și P_{ref} pentru care vectorul \mathbf{n} al reperului de vizualizare este $\mathbf{n} = (0, 1, 0)$ sunt $P_0 =$, $P_{\text{ref}} =$
14. (2p) Un exemplu de alegere a parametrilor funcției `glm::ortho` pentru care paralelipipedul decupat să aibă volumul egal cu 36 este
15. (2p) O sursă de lumină amplasată în punctul de coordonate omogene $[a : b : c : d]$ (**alegeți a, b, c, d**) este o sursă

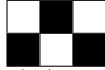
II. (15p) Rezolvați complet problemele.

16. (5p) Alegeți un dreptunghi $ABCD$ situat într-un plan de forma $y = \text{constant}$ (i.e. paralel cu planul Oxz). Indicați un punct M situat în fața dreptunghiului $ABCD$ și un punct N situat în spatele dreptunghiului $ABCD$. Justificați! Stabiliți dacă segmentul $[MN]$ intersectează dreptunghiul în interior, pe laturi sau în exterior.

17. (5p) a) Este aplicată secvența de cod sursă

```
glm::translate(t1, t2, 0.0);  
glm::scale(s1, s2, 0.0);
```

dreptunghiului delimitat de dreptele $x = \alpha$, $x = \beta$, $y = \gamma$, $y = \theta$. Alegeți valori pentru $\alpha, \beta, \gamma, \theta$ astfel ca dreptunghiul să fie nevid. Alegeți apoi valori nenule pentru t_1, t_2, s_1, s_2 ; $s_1 \neq 1$, $s_2 \neq 1$, $s_1 \neq s_2$, astfel ca în urma aplicării acestei secvențe de cod sursă să fie obținut un pătrat cu centrul în origine. Justificați!

b) Pentru alegerile făcute, aplicăm pătratului rezultat în urma transformării texturii ; coordonatele de texturare asociate vârfurilor pătratului sunt $(0, 0)$ (stânga jos), $(5, 0)$ (dreapta jos), $(5, 4)$ (dreapta sus), $(0, 4)$ (stânga sus), iar fundalul este roșu. Stabiliți care este raportul dintre aria colorată cu alb și cea colorată cu negru, știind că este utilizată opțiunea `GL_CLAMP`.

18. (5p) Pentru implementarea modelului de iluminare este utilizată formula care ține cont de componenta ambientală și de reflexia difuză, neglijând reflexia speculară, conform codului de mai jos:

```
vec3 ambient_light = ambientStrength * lightColor; // #1  
vec3 ambient_term = ambient_light * objectColor; // #2  
vec3 norm = normalize(Normal); // #3  
vec3 lightDir = normalize(lightPos - FragPos); // #4  
float diff = max(dot(norm, lightDir), 0.0); // #5  
vec3 diffuse_light = lightColor; // #6  
vec3 diffuse_term = diff * diffuse_light * objectColor; // #7  
vec3 result = ambient_term + diffuse_term; // #8  
out_Color = vec4(result, 1.0f); // #9
```

Se consideră un fragment de coordonate $(3, 1, 4)$ cu normala dată de $(0, 1, 0)$ și cu `objectColor = (0.3, 0.2, 0.5)`. Sursa de lumină de lumină este situată în punctul $(3, 4, 4)$. Stabiliți ce alte elemente mai trebuie indicate, alegeți niște valori adecvate nenule și calculați valoarea culorii pentru fragmentul respectiv, explicând pentru fiecare linie calculul efectuat (eticheta liniei i este indicată la final, sub forma **#i**).