

I. (25p) Indicați răspunsul corect.

1. (1p) Un cod RGB care să genereze un ton de albastru închis este
2. (1p) O funcție utilizată în shader care permite “comunicarea” cu programul principal este
3. (1p) O funcție care are ca efect schimbarea sensului de parcurgere a unui poligon este
4. (1p) O funcție care permite explicit desenarea unei primitive este
5. (1p) Pentru a putea decupa și reprezenta o scenă 3D 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-au obținut $\Delta x = \dots$, $\Delta y = \dots$ (alegeți valorile, astfel ca panta $m \in (0, 1)$), atunci valoarea lui p_0 este
8. (2p) Date punctele $A = (0, 1, 1)$, $B = (0, 2, 2)$, $C = (\alpha, \beta, \gamma)$ (alegeți valorile, astfel ca C să nu fie situat pe dreapta AB), un punct situat în exteriorul triunghiului ABC este
9. (2p) Folosind regula par-impair pentru o linie poligonală închisă, un punct interior este obținut pentru valorile $n_+ = \dots$, $n_- = \dots$ (alegeți valori).
10. (2p) În planul proiectiv $\mathbb{P}^2\mathbb{R}$ punctul $[-4 : \alpha : 1]$ (alegeți $\alpha \neq 0$) coincide cu punctul (alegeți, α).
11. (2p) Produsul elementelor nenule ale matricei 4×4 asociate translației de vector $\mathbf{v} = \dots$ (alegeți un vector nenul $\mathbf{v} \in \mathbb{R}^3$) este
12. (2p) Un exemplu de aplicație afină $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ pentru care $f(0, 0) = (0, 0)$ și $f(0, 1) = (a, b)$ (alegeți valorile a, b este
13. (2p) Se utilizează `glm::lookAt(1, 2, 3, 4, 5, 6, 0, 1, 0)`. Vectorul n al reperului de vizualizare este
14. (2p) Un exemplu de alegere a parametrilor funcției `glm::ortho` pentru care paralelipipedul decupat are volumul egal cu 36 este
15. (2p) Avem o sursă de lumină în $L = (4, 6, 4)$, iar proiecția se realizează pe planul $z = 2$. Fie \mathcal{P} reuniunea dintre patrulaterul $ABCD$ și interiorul său (cu $A = (2, 5, 3)$, $B = (6, 5, 3)$, $C = (6, 7, 3)$, $D = (2, 7, 3)$). Un punct din planul $z = 2$ care este situat în umbra lui \mathcal{P} este


II. (15p) Rezolvați complet problemele.

16. (5p) Alegeți patru puncte A, B, C, D situate în planul $y = 2$ și care să determine un pătrat. Stabiliți unde este situat punctul $O = (0, 0, 0)$ în raport cu pătratul $ABCD$ (față sau spate).

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

```
glm::translate (30.0, -10.0, 0.0);  
glm::scale (0.4, 2.0, 0.0);  
glm::translate (-10.0, 20.0, 0.0);
```

dreptunghiului delimitat de dreptele $x = a, x = b, y = c, y = d$. Alegeți valori pentru a, b, c, d astfel ca dreptunghiul să fie nevid și inclus în cadranul I, apoi stabiliți care este poziția centrului dreptunghiului după aplicarea secvenței de cod sursă de mai sus.

b) Pentru alegerile făcute, aplicăm pătratului rezultat în urma transformării textura ; coordonatele de texturare asociate vârfurilor pătratului sunt $(0, 0)$ (stânga jos), $(4, 0)$ (dreapta jos), $(4, 2)$ (dreapta sus), $(0, 2)$ (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 $(3, -2, 4)$ cu normala dată de $(0, 1, 0)$. Sursa de lumină este situată în punctul $(3, 2, 6)$ și are `lightColor`=(0.6, 0.2, 0.1). 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 $\#i$).