NATIONAL UNIVERSITY OF SINGAPORE

**CS3241 — Computer Graphics**

AY2023/2024 Semester 1

**Midterm Assessment**

Time Allowed: 1 Hour 30 Minutes

# SOLUTIONS

---

**INSTRUCTIONS**

1. This **QUESTION PAPER** contains **27 Multiple-Choice Questions** (MCQs) in **4 Sections**, and comprises **10** printed pages, including this page.
2. The **ANSWER SHEET** comprises **2** printed pages.
3. Use a pen or pencil to **write** your **Student Number** in the designated space on the front page of the **ANSWER SHEET**, and **shade** the corresponding circle **completely** in the grid for each digit or letter. DO NOT WRITE YOUR NAME!
4. You must **submit only** the **ANSWER SHEET** and no other documents. Do not tear off any pages from the ANSWER SHEET.
5. All questions must be answered in the space provided in the **ANSWER SHEET**; no extra sheets will be accepted as answers.
6. Write legibly with a **pen** or **pencil** (do not use red color). Untidiness will be penalized.
7. For **multiple choice questions (MCQ)**, **shade** in the **circle** of the correct answer **completely**. Each question has one correct answer. The **indicated marks** are awarded for each correct answer and there is **no penalty** for a wrong answer.
8. The full score of this assessment is **80** marks.
9. This is an **Open-Book** assessment.
10. You are allowed to use an approved **calculator**.

# Section A [17 marks]

Let ***S*** be a **sphere** of unit radius and centered at the origin. ***S*** is approximated and represented as a mesh of **1224** triangles. All vertices of the triangles lie on the surface of ***S***.

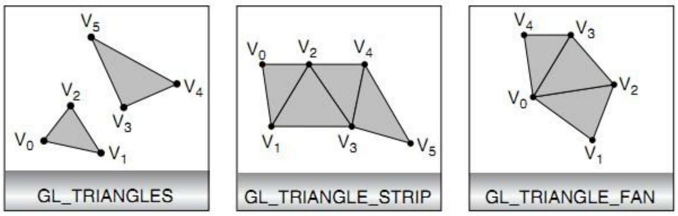
**(1)** **[3 marks]** How many ***distinct* vertices** are there in the representation of the sphere?

**A.** 612
**B.** 614 **(answer)**
**C.** 1220
**D.** 1224
**E.** 2444
**F.** 2448

**(2)** **[2 marks]** Suppose we want to draw the sphere using the `GL_TRIANGLES` OpenGL primitive mode, how many times do we need to call the `glVertex*()` function?

**A.** 408
**B.** 612
**C.** 1224
**D.** 1227
**E.** 2448
**F.** 3672 **(answer)**

**(3)** **[3 marks]** Suppose all the triangles of the sphere can be drawn as two **triangle fans** and 16 **triangle strips**. Each triangle strip consists of 72 triangles. How many times do we need to call the `glVertex*()` function to draw **one triangle strip**?




**A.** 24
**B.** 36
**C.** 38
**D.** 74 **(answer)**
**E.** 72
**F.** 146

**(4)** **[3 marks]** Let $\boldsymbol{v}$ be a triangle vertex on the sphere, and it has coordinates $(x_0, y_0, z_0)$. Which of the following is a **normal vector** of the sphere $\boldsymbol{S}$ at vertex $\boldsymbol{v}$? Note that the normal vector need not be a unit vector, and it must be pointing towards the "outside" of the sphere.

**A.** $(x_0, y_0, z_0)$ **(answer)**
**B.** $(-x_0, -y_0, -z_0)$
**C.** $(-y_0, z_0, -x_0)$
**D.** (1, 0, 0)
**E.** (0, 1, 0)
**F.** (0, 0, 1)

**(5)** **[2 marks]** Suppose the scene we are rendering has nothing else besides the sphere $\boldsymbol{S}$, and we have set up a perspective view volume that encloses sphere $\boldsymbol{S}$ entirely. If we **turn on** z-buffering (and properly clear it before rendering each frame) and **turn off** backface culling, will the sphere $\boldsymbol{S}$ be rendered with correct hidden surface removal?

**A.** Yes **(answer)**
**B.** No

**(6)** **[2 marks]** Suppose the scene we are rendering has nothing else besides the sphere $\boldsymbol{S}$, and we have set up a perspective view volume that encloses sphere $\boldsymbol{S}$ entirely. If we **turn off** z-buffering and **turn on** backface culling, will the sphere $\boldsymbol{S}$ be rendered with correct hidden surface removal?

**A.** Yes **(answer)**
**B.** No

**(7)** **[2 marks]** Suppose the scene we are rendering has nothing else besides the sphere $\boldsymbol{S}$, and we have set up a perspective view volume that encloses sphere $\boldsymbol{S}$ entirely. If we **turn off** z-buffering and **turn off** backface culling, will the sphere $\boldsymbol{S}$ be rendered with correct hidden surface removal?

**A.** Yes
**B.** No **(answer)**

# Section B [9 marks]

**(8)** **[3 marks]** Suppose we have 3 pieces of glass filters of equal thickness, of which one is **cyan** color, one is **magenta**, and one is **yellow**. We want to shine a beam of white light through the glass filters to produce **blue light**. Which is/are the glass filter(s) that we should shine the white light through?

**A.** Magenta only
**B.** Yellow only
**C.** Magenta and yellow
**D.** Yellow and cyan
**E.** Cyan and magenta **(answer)**
**F.** Cyan, magenta and yellow

**(9)** **[3 marks]** Let $F_1$ be a flat piece of colored glass that lets 80% of red light, 60% of green light, and 40% of blue light, pass through. Another flat piece of glass, $F_2$, lets 90% of red light, 30% of green light, and 40% of blue light, pass through. If we shine a beam of white light through glass $F_1$ and glass $F_2$, in that order, how much of the white light actually passes through both $F_1$ and $F_2$?

**A.** 28% of red, 82% of green, 84% of blue
**B.** 80% of red, 60% of green, 40% of blue
**C.** 72% of red, 18% of green, 16% of blue **(answer)**
**D.** 90% of red, 30% of green, 40% of blue
**E.** 2% of red, 28% of green, 36% of blue
**F.** 18% of red, 12% of green, 24% of blue

**(10)** **[3 marks]** A beam of white light is shining through an alternating glass filter that is changing color between **yellow** and **cyan** at a rate of 240 Hz. Suppose the filtered light beam falls on a white matte opaque surface, what will be the perceived color on the white surface?

**A.** Light red
**B.** Dark blue
**C.** Dark green
**D.** Light green **(answer)**
**E.** Magenta
**F.** Red

# Section C [27 marks]

**(11)** **[2 marks]** What does the homogeneous coordinates $[8\ \ 6\ \ 4\ \ 0]^T$ represent?

- **A.** The 3D point (8, 6, 4)
- **B.** The 3D point (4, 3, 2)
- **C.** The 3D point (1/8, 1/6, 1/4)
- **D.** The 3D vector (8, 6, 4) **(answer)**
- **E.** The 3D vector (4, 3, 2)
- **F.** The 3D vector (1/8, 1/6, 1/4)

**(12)** **[2 marks]** What does the homogeneous coordinates $[6\ \ 4\ \ 2\ \ 0.5]^T$ represent?

- **A.** The 3D point (3, 2, 1)
- **B.** The 3D point (6, 4, 2)
- **C.** The 3D point (12, 8, 4) **(answer)**
- **D.** The 3D point (1/12, 1/8, 1/4)
- **E.** The 3D vector (6, 4, 2)
- **F.** The 3D vector (12, 8, 4)

**(13)** **[3 marks]** Which of the following is the matrix that rotates objects about the point (8, 5, 7), where the rotation axis is the vector (0, 0, 1), and the rotation angle is θ? Note that $\mathbf{T}(d_x, d_y, d_z)$ is a translation matrix for displacing a point by $(d_x, d_y, d_z)$, and $\mathbf{R}_z(\alpha)$ is a rotation matrix for rotating a point about the *z*-axis by an angle of $\alpha$.

- **A.** $\mathbf{T}(8, 5, 0) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{T}(-8, -5, 0)$ **(answer)**
- **B.** $\mathbf{T}(-8, -5, 0) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{T}(8, 5, 0)$
- **C.** $\mathbf{T}(8, 0, 7) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{T}(-8, 0, -7)$
- **D.** $\mathbf{T}(-8, 0, -7) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{T}(8, 0, 7)$
- **E.** $\mathbf{T}(0, 5, 7) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{T}(0, -5, -7)$
- **F.** $\mathbf{T}(0, -5, -7) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{T}(0, 5, 7)$

**(14)** **[4 marks]** Suppose the **model-view matrix** is a 4×4 scaling matrix **S**(2, 3, 4), where 2, 3, and 4 are the scaling factors for the *x*, *y*, and *z* directions, respectively. Which of the following is the corresponding 3×3 **matrix** used for **transforming normal vectors**? The function *mat3x3* extracts and returns the upper-left 3×3 submatrix of its input 4×4 matrix.

**A.** *mat3x3*( **S**(2, 3, 4) )
**B.** *mat3x3*( **S**(1/2, 1/3, 1/4) ) **(answer)**
**C.** *mat3x3*( **S**(–2, –3, –4) )
**D.** *mat3x3*( **S**(–1/2, –1/3, –1/4) )
**E.** *mat3x3*( **S**(4, 3, 2) )
**F.** *mat3x3*( **S**(1/4, 1/3, 1/2) )

**(15)** **[4 marks]** Consider the following view transformation matrix:

$$\mathbf{M}_{\text{view}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -7 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Which of the following `gluLookAt()` function calls produces the above view transformation matrix?

**A.** `gluLookAt( 7, 3, 5,  7, 2, 5,  1, 0, 0 );`
**B.** `gluLookAt( 7, 3, 5,  6, 3, 5,  1, 0, 0 );`
**C.** `gluLookAt( 7, 3, 5,  6, 3, 5,  0, 1, 0 );`
**D.** `gluLookAt( 7, 3, 5,  8, 3, 5,  0, 1, 0 );`
**E.** `gluLookAt( 7, 3, 5,  7, 2, 5,  0, 0, 1 );`
**F.** `gluLookAt( 7, 3, 5,  6, 3, 5,  0, 0, 1 );` **(answer)**

**(16)** **[3 marks]** A vertex, whose **camera coordinates** are (–100, 50, –250), is being projected using the following OpenGL orthographic projection:

```
glOrtho( -200, 200, 0, 100, 100, 300 );
```

What will be the vertex's Normalized Device Coordinates (NDC)?

**A.** (–0.5, 0.5, 0.5)
**B.** (–0.5, 0.5, –0.5)
**C.** (0.25, 0.5, 0.75)
**D.** (0.25, 0.5, 0.25)
**E.** (–0.5, 0, 0.5) **(answer)**
**F.** (–0.5, 0, –0.5)

**(17)** **[3 marks]** Suppose the **viewport** is set up as `glViewport(50, 80, 800, 600)`, and the entire viewport is within the rendering window, what are the **2D window coordinates** and **depth value** of the NDC point (0, 0, 0.5)?

**A.** Window coordinates: (50, 80), and depth: 0.25
**B.** Window coordinates: (50, 80), and depth: 0.50
**C.** Window coordinates: (50, 80), and depth: 0.75
**D.** Window coordinates: (450, 380), and depth: 0.25
**E.** Window coordinates: (450, 380), and depth: 0.50
**F.** Window coordinates: (450, 380), and depth: 0.75 **(answer)**

**(18)** **[3 marks]** Suppose two diagonally-opposite corners on the near plane of a **perspective viewing volume** are at (12, 6, –10) and (42, 36, –10) in the camera space, and the far plane distance is 20, which of the following `glFrustum` function calls correctly sets up the viewing volume?

**A.** `glFrustum(-12, 42, -6, 36, 10, 20);`
**B.** `glFrustum(12, 42, 6, 36, 10, 20);` **(answer)**
**C.** `glFrustum(6, 21, 3, 18, 10, 20);`
**D.** `glFrustum(-6, 21, -3, 18, 10, 20);`
**E.** `glFrustum(12, 42, 6, 36, -10, -20);`
**F.** `glFrustum(6, 21, 3, 18, -10, -20);`

**(19)** **[3 marks]** Which of the following view volume settings is the most prone to **z-fighting**?

**A.** `gluPerspective(100, 1, 1, 20);` **(answer)**
**B.** `gluPerspective(20, 1, 1, 20);`
**C.** `gluPerspective(100, 1, 201, 220);`
**D.** `gluPerspective(20, 1, 201, 220);`
**E.** `gluPerspective(100, 1, 101, 120);`
**F.** `gluPerspective(20, 1, 101, 120);`

# Section D [27 marks]

**(20)** **[2 marks]** Consider using the Cohen-Sutherland Algorithm to clip a **2D line segment** against a **2D rectangular clipping window**. The diagonally opposite corners of the clipping window are at (200, 100) and (500, 300). What is the outcode of a line segment endpoint located at (700, 50)?

**A.** `1010`
**B.** `0110` **(answer)**
**C.** `1001`
**D.** `0101`
**E.** `0100`
**F.** `0010`

**(21)** **[3 marks]** Consider using the Cohen-Sutherland Algorithm to clip a **3D line segment** against a **3D axis-aligned rectangular clipping box**. Suppose the two endpoints have outcodes `100110` and `010110`, what is the maximum number of **line-plane intersections** that need to be computed?

**A.** 0 **(answer)**
**B.** 1
**C.** 2
**D.** 3
**E.** 4
**F.** 5

**(22)** **[3 marks]** Consider using the Cohen-Sutherland Algorithm to clip a **3D line segment** against a **3D axis-aligned rectangular clipping box**. Suppose the two endpoints have outcodes `101010` and `010100`, what is the maximum number of **line-plane intersections** that need to be computed?

**A.** 0
**B.** 1
**C.** 2
**D.** 3
**E.** 4
**F.** 5 **(answer)**

**(23)** **[3 marks]** Pixel $P$ in the framebuffer is covered by triangles $T_1$, $T_2$, $T_3$, $T_4$, $T_5$, $T_6$, $T_7$, and $T_8$ only. The triangles are rendered in the order $T_1$, $T_2$, …, $T_8$. Suppose the **depth values** of the fragments of $T_1$, $T_2$, …, $T_8$ at the location of $P$ are 0.90, 0.60, 0.80, 0.70, 0.20, 0.10, 0.50, 0.20, respectively, how many times will the z-value in the **z-buffer** for pixel $P$ be modified *after* it was initialized?

**A.** 7
**B.** 6
**C.** 5
**D.** 4 **(answer)**
**E.** 3
**F.** 2

**(24)** **[3 marks]** A **line segment** from the pixel location (200, 100) to (100, 400) is **scan-converted**. Assuming the line segment is drawn as thin as possible and not broken, what is the number of fragments (pixels that are turned on) that are produced for this line segment? This number includes the two fragments at the two endpoints of the line segment.

**A.** 101
**B.** 100
**C.** 301 **(answer)**
**D.** 300
**E.** 401
**F.** 400

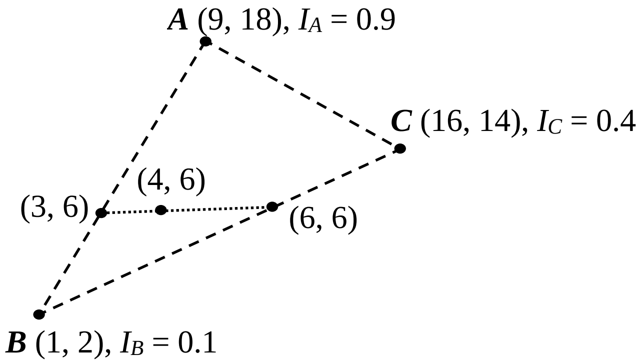
**(25)** **[5 marks]** Consider that the center of each pixel has integer *x*-coordinate and *y*-coordinate. We want to scan convert a **45-degree arc** of a circle of **radius 58** and centered at (100, 100). The arc starts at pixel location (100, 158) and extends clockwise to the other end point exactly 45 degrees away. Assuming the curve is drawn as thin as possible and not broken, what is the **number of fragments** (pixels that are turned on) that are produced for this arc? This number includes the two fragments at the two end points of the curve. (Note: $\cos 45° = 1/\sqrt{2} \approx 0.707107$)

**A.** 29
**B.** 30
**C.** 59
**D.** 58
**E.** 42 **(answer)**
**F.** 41

**(26)** **[3 marks]** Now, we want to scan convert the **entire circle**, which has a **radius of 58** and is centered at (100, 100). Let $F$ be the correct number of fragments produced for the 45-degree arc in the preceding question, what is the **number of fragments** for the full circle?

**A.** $8F - 8$ **(answer)**
**B.** $8F - 4$
**C.** $8F$
**D.** $8F + 4$
**E.** $8F + 8$
**F.** $4F - 4$

**(27)** **[5 marks]** Given the following triangle whose vertices ***A***, ***B***, ***C*** are at *pixel locations* (9, 18), (1, 2), and (16, 14) respectively, and the **intensity** attribute values at the vertices are 0.9, 0.1, 0.4 respectively. (Note that the diagram is not drawn to scale.) What is the intensity value at the pixel location (4, 6) assuming the triangle is rasterized with smooth shading?




**A.** 0.300
**B.** 0.333
**C.** 0.200
**D.** 0.233
**E.** 0.250
**F.** 0.267 **(answer)**

**——— END OF QUESTION PAPER ———**