Student Number: **Name:**

Write down answers in-between questions. Please answer using short sentences.

The back of each page can be used for practice, but DO NOT write down the answer on the back.

Be sure to write your student number and name on each page.

1. (3 pts) ____(a)____ is the loss of one degree of freedom in Euler angle representation for 3D rotation that occurs when the two axes of three gimbals are aligned. Fill in the blank (a).

2. (6 pts) Write down the lowest-degree polynomial x(t) that passes through 3 data points x(0)=1, x(1)=-2, and x(2)=3.

3. (6 pts) Choose **ALL false(incorrect)** statements about the visibility problem.
   1) Invisible primitives should be removed for efficient and correct rendering.
   2) Clipping is the process of removing primitives occluded by other objects closer to the camera.
   3) Back-face culling uses the dot product between surface normal and camera view vector.
   4) Painter's algorithm is the most popular hidden surface removal algorithm.

4. (6 pts) Choose **ALL false(incorrect)** statement about the Phong illumination model.
   1) Phong illumination model is composed of three components; ambient, diffuse, specular component.
   2) The ambient component provides constant global lighting, which is a severe approximation of indirect lighting.
   3) The diffuse component assumes that light is reflected equally in all directions.
   4) The intensity of the diffuse component increases as the angle between the surface normal vector and the eye direction vector decreases.
   5) The specular component simulates highlights on shiny objects in a physically correct way.
   6) The intensity of the specular component increases as the angle between the surface normal vector and the light direction vector decreases.

Student Number: **Name:**

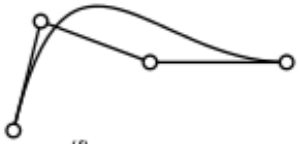
5. (6 pts) Choose **ALL impossible** combinations for Euler angles.
   1) XYZ
   2) ZXZ
   3) ZXX
   4) YZX
   5) YYX
   6) YXY

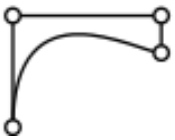
6. (6 pts) Choose **ALL false(incorrect)** statements about the interpolation of 3D orientations (or rotations). (Note that $\text{slerp}(\mathbf{R}_1, \mathbf{R}_2, t) = \mathbf{R}_1 \exp(t \cdot \log(\mathbf{R}_1^T \mathbf{R}_2))$)
   1) Slerp is a correct method for linear interpolation of two 3D orientations.
   2) Linear interpolation between two rotation vectors results in correctly interpolated rotation as well.
   3) In terms of rotation vectors and rotation matrixes, logarithm mapping (log()) means converting rotation matrix representation to rotation vector representation.
   4) The Rodrigues' rotation formula can be an implementation of logarithm mapping (log()).

7. (6 pts) Below are four curves and their "control points/polygon." Some of the control polygons are the Bezier control polygon for the curve drawn with it; the others are not. Choose **ALL Bezier control polygon**. You may assume that none of the control points overlap or are repeated.

a. 

b. 

c. 

d. 

8. (12 pts) Below is the pseudo code for z-buffer (depth buffer) algorithm. Fill in the blanks (a), (b), and (c). You have to use functions already used in the code and variables already defined in the code. Additionally, you can use a function call `read_depth_buffer(x,y)` to get the current recorded depth value at (x,y).

Student Number: **Name:**

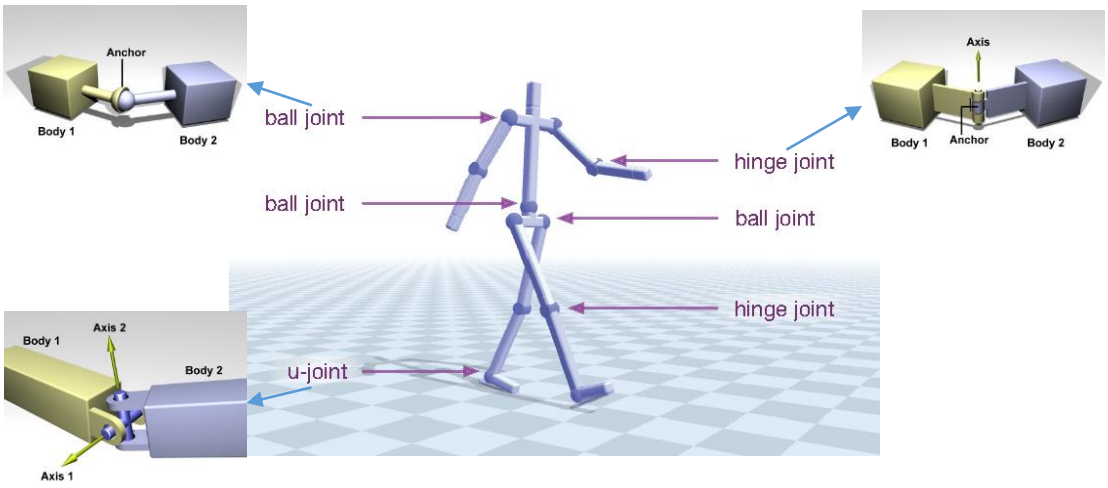
```
allocate depth_buffer;

for each pixel (x,y)
    write_frame_buffer(x,y,backgrnd_color);
    write_depth_buffer(x,y,farPlane_depth);

    for each polygon
        for each pixel (x,y) in polygon
            new_color = polygon's color at (x,y);
            new_depth = polygon's z-value at (x,y);
            if (_____(a)_____)
                _____(b)_____;
                _____(c)_____;
```

9. (8 pts) Show that $\mathbf{R_1R_2}$ is a rotation matrix if $\mathbf{R_1}$ and $\mathbf{R_2}$ are rotation matrices. Use the two major mathematical properties of a rotation matrix.

10. (3 pts) What is the total degrees of freedom of the following system? Note that the root joint can move freely.



Student Number: **Name:**

11. (8 pts) The following render() function renders the scene with incorrect lighting. What modification is required for this code to do lighting correctly? Assume that drawFrame() and drawCube_glDrawArray() are implemented correctly and use of numpy module as np, global variables, and gluLookAt() does not have any problems.

```
def render():
    global gCamAng, gCamHeight
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)

    glEnable(GL_DEPTH_TEST)

    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(45, 1, 1,10)

    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    gluLookAt(5*np.sin(gCamAng),gCamHeight,5*np.cos(gCamAng), 0,0,0,
0,1,0)

    drawFrame()

    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)

    glPushMatrix()
    lightPos = (3.,4.,5.,1.)
    glLightfv(GL_LIGHT0, GL_POSITION, lightPos)
    glPopMatrix()

    lightColor = (1.,1.,1.,1.)
    ambientLightColor = (.1,.1,.1,1.)
    glLightfv(GL_LIGHT0, GL_DIFFUSE, lightColor)
    glLightfv(GL_LIGHT0, GL_SPECULAR, lightColor)
    glLightfv(GL_LIGHT0, GL_AMBIENT, ambientLightColor)

    objectColor = (1.,0.,0.,1.)
    specularObjectColor = (1.,1.,1.,1.)
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, objectColor)
    glMaterialfv(GL_FRONT, GL_SHININESS, 10)
    glMaterialfv(GL_FRONT, GL_SPECULAR, specularObjectColor)

    glPushMatrix()
    glScalef(.2,.2,.2)
    drawCube_glDrawArray()
    glPopMatrix()

    glDisable(GL_LIGHTING)
```

Student Number: **Name:**

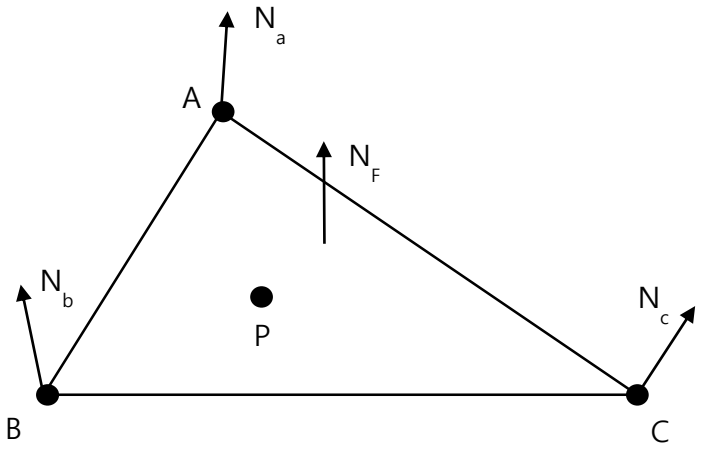
12. (15 pts) The following text is the hierarchy section of a bvh file which has 6 joints. If each joint's position w.r.t. global frame is given as the following table, fill in the blanks (a), (b), (c), (d), and (e).

joint	global position
J0	(0.0, 0.0, 0.0)
J1	(0.5, 0.5, 0.0)
J2	(0.75, 1.0, 0.0)
J3	(0.25, 1.0, 0.0)
J4	(-0.5, 0.5, 0.0)
J5	(-0.5, 1.0, 0.0)

```
HIERARCHY
ROOT J0
{
	OFFSET 0.00   0.00   0.00
	CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation
Yrotation
	JOINT J1
	{
		OFFSET  ____(a)____
		CHANNELS 3 Zrotation Xrotation Yrotation
		JOINT J2
		{
			OFFSET  ____(b)____
			CHANNELS 3 Zrotation Xrotation Yrotation
			End Site
			{
				OFFSET  0.00   0.50   0.00
			}
		}
		JOINT J3
		{
			OFFSET  ____(c)____
			CHANNELS 3 Zrotation Xrotation Yrotation
			End Site
			{
				OFFSET  0.00   0.50   0.00
			}
		}
	}
	JOINT J4
	{
		OFFSET  ____(d)____
		CHANNELS 3 Zrotation Xrotation Yrotation
		JOINT J5
		{
			OFFSET  ____(e)____
			CHANNELS 3 Zrotation Xrotation Yrotation
			End Site
			{
				OFFSET  0.00   0.50   0.00
			}
		}
	}
}
```

Student Number: **Name:**

13. (15 pts) You're about to render a triangle ABC which is one of triangles in a triangle mesh model. N_a, N_b, N_c are the vertex normal of the vertex A, B, C and calculated as the average of the face normals of the neighboring triangles. N_F is the face normal of triangle ABC.



The illumination model used for this rendering is given as a function `illum(p, n)` where p is the position of the target point to be illuminated and n is the surface normal at p to be used for illumination. For example, the illuminated color at the vertex A with N_a as surface normal is `illum(A, Na)`.

Question: What is the illuminated color at the point P in the triangle ABC when using…

1) Flat shading
2) Gouraud shading
3) Phong shading

Note that the barycentric coordinate of the interior point P is given as (u, v, w), meaning that P = uA + vB + wC. You have to use u, v, w to compute interpolated vertex color or vertex normal at P.

Note that A, B, C, and P mean both the name of vertices (or points) and their positions.