CompSci 372 S1 C 2003
Computer Graphics

Mid Term Test 2nd May 2003, 6.30 pm – 7.30 pm

Surname (Family Name):	
First Name(s):	
ID Number:	
Login Name (UPI):	

Instructions:

1. Attempt **ALL** questions.
2. The test is for one (1) hour.
3. This is a closed book test.
4. Calculators are **NOT** permitted.
5. Write your answers in the spaces provided. There is space at the back for answers that overflow the allotted space.
6. **Questions total 50 Marks**.
7. This test is worth 10% of your final marks for CompSci372 S1 C

Section	Marks	Maximum Marks
Q.1		11
Q.2		10
Q.3		11
Q.4		12
Q.5		6
Total		50

Question 1 – Short answer test [11 marks]

Complete each of the following statements by filling in the underlined blank spaces. Each blank space is worth 1 mark.

[11 marks]

(a) Modern PC graphics card have a large memory called the **VRAM / frame buffer** which is constantly scanned by the **CRT controller / video controller** to get a video signal.

(b) Currently there are two popular APIs used for hardware supported 3D graphics: **DirectX/Direct3D** is heavily promoted by Microsoft but **OpenGL** is an open API which exists for all major platforms.

(c) If **v** = (1,-1,4) and **u** = (-2,3,3) then the dot product is **v.u** = **-2-3+12=7** .

(d) Let α be a real number. Then $cos^2\,\alpha + sin^2\,\alpha =$ **1** .

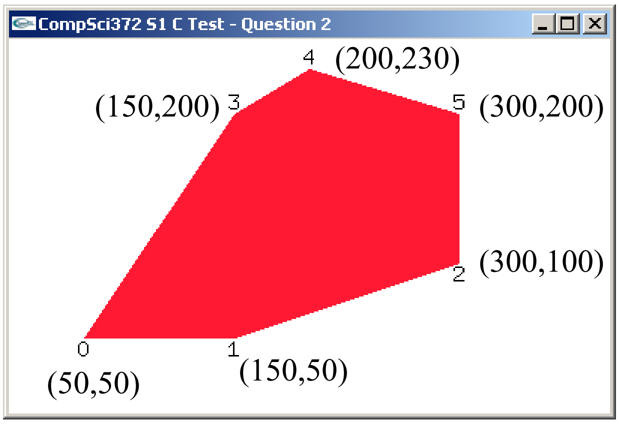
(e) Given a parametric surface **p**(s,t) with 0≤s,t≤1 the tangent in s-direction is $\frac{\partial \mathbf{p}}{\partial s}$.

(f) The parametric equation for a line segment from point $\mathbf{p}_0$ to $\mathbf{p}_1$ is $\mathbf{p}(t) = \mathbf{p}_0 + t\,(\mathbf{p}_1 - \mathbf{p}_0)$ where $t \in$ **[0,1]** .

The command glMatrixMode(GL_MODELVIEW) selects a **matrix stack** which handles both modelling operations and **view transformations**.

Question 2 – OpenGL [10 marks]

In this question you have to draw the 2D object shown on the right. The vertices 0 to 5 of the object and the corresponding coordinates are shown in the image on the right.



A. Complete the code fragment below so that it defines a global array containing the six 2D vertices of the above shape [2 marks]:

```
const int numVertices=6;
const float v[numVertices][2] = {{50, 50}, {150, 50}, {300, 100}, {150, 200},
                                  {200, 230}, {300,200}};
```

B. Complete the display function below so that it draws the shape defined by these six vertices using the **GL_QUADS** mode and the **glVertex2fv** command. The vertex numbers in the resulting image (shown above) have been inserted afterwards for clarity and you don't have to draw them. [3 marks]

```
void display(void)
{
    // clear all pixels in frame buffer
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.1, 0.2);    // reddish colour
    glBegin(GL_QUADS);

    glVertex2fv(v[0]);
    glVertex2fv(v[1]);
    glVertex2fv(v[2]);
    glVertex2fv(v[3]);
    glVertex2fv(v[3]);
    glVertex2fv(v[2]);
    glVertex2fv(v[5]);
    glVertex2fv(v[4]);

    glEnd();
    glFlush();
}
```

C. Complete the display function below so that it draws the shape defined by these six vertices using the **GL_QUAD_STRIP** mode and the **glVertex2fv** command [3 marks].

```
void display(void)
{
    // clear all pixels in frame buffer
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.1, 0.2);    // reddish colour
    glBegin(GL_QUAD_STRIP);

    glVertex2fv(v[0]);
    glVertex2fv(v[3]);
    glVertex2fv(v[1]);
    glVertex2fv(v[4]);
    glVertex2fv(v[2]);
    glVertex2fv(v[5]);

    glEnd();
    glFlush();
}
```

D. Complete the display function below so that it draws the shape defined by these six vertices using the **GL_TRIANGLE_FAN** mode and the **glVertex2fv** command [2 marks].

```
void display(void)
{
    // clear all pixels in frame buffer
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.1, 0.2);    // reddish colour
    glBegin(GL_TRIANGLE_FAN);

    glVertex2fv(v[0]);
    glVertex2fv(v[1]);
    glVertex2fv(v[2]);
    glVertex2fv(v[5]);
    glVertex2fv(v[4]);
    glVertex2fv(v[3]);

    glEnd();
    glFlush();
}
```

Question 3 – 3D Geometry [11 marks]

A. Find the equation **n•p**=*d* of the plane defined by the three points

$$\mathbf{p}_1 = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} 4 \\ 2 \\ 3 \end{pmatrix}, \quad \mathbf{p}_3 = \begin{pmatrix} 1 \\ 4 \\ -1 \end{pmatrix}$$

[5 marks]

$$\mathbf{n} = (\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1) = \begin{pmatrix} 3 \\ 0 \\ 4 \end{pmatrix} \times \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0*0-4*2 \\ 4*0-0*3 \\ 3*2-0*0 \end{pmatrix} = \begin{pmatrix} -8 \\ 0 \\ 6 \end{pmatrix}$$

$$d = \mathbf{n} \bullet \mathbf{p}_1 = \begin{pmatrix} -8 \\ 0 \\ 6 \end{pmatrix} \bullet \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} = -14$$

Plane equation is: $\begin{pmatrix} -8 \\ 0 \\ 6 \end{pmatrix} \bullet \mathbf{p} = -14$

B. Derive an equation for the distance d of a 3D line $\mathbf{p}(t)=\mathbf{p}_0+t\mathbf{v}$ from the origin.
[Hint: The distance of the line from the origin is the minimum distance of all points on the line to the origin. The vector from the origin to the point on the line closest to it is orthogonal to the line.]

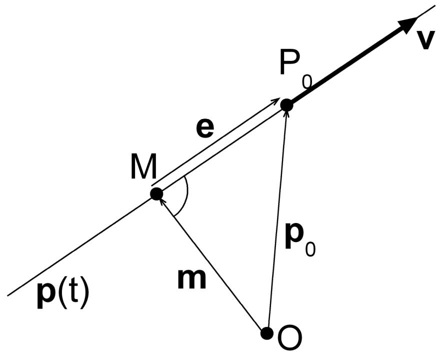
[6 marks]

Solution 1:

$$\mathbf{e} = \frac{\mathbf{p}_0 \bullet \mathbf{v}}{\mathbf{v} \bullet \mathbf{v}} \mathbf{v}$$

$$\mathbf{m} = \mathbf{p}_0 - \mathbf{e}$$

$$d = |\mathbf{m}| = \left| \mathbf{p}_0 - \frac{\mathbf{p}_0 \bullet \mathbf{v}}{\mathbf{v} \bullet \mathbf{v}} \mathbf{v} \right|$$



Solution 2: Find $\hat{t}$ such that $\mathbf{p}(\hat{t}) = \mathbf{m}$

$$\mathbf{p}(\hat{t}) \perp \mathbf{v} \Leftrightarrow \mathbf{p}(\hat{t}) \bullet \mathbf{v} = 0$$

$$\Leftrightarrow \left(p_0 + \hat{t}\mathbf{v} \right) \bullet \mathbf{v} = 0$$

$$\Leftrightarrow p_0 \bullet \mathbf{v} + \hat{t}\mathbf{v} \bullet \mathbf{v} = 0$$

$$\Leftrightarrow \hat{t} = -\frac{p_0 \bullet \mathbf{v}}{\mathbf{v} \bullet \mathbf{v}}$$

$$d = |\mathbf{m}| = \left| p_0 - \frac{p_0 \bullet \mathbf{v}}{\mathbf{v} \bullet \mathbf{v}} \mathbf{v} \right|$$

Solution 3: Find $\hat{t}$ such that $\left|\mathbf{p}(\hat{t})\right|$ is minimal

$$\left|\mathbf{p}(\hat{t})\right| = \sqrt{\mathbf{p}(\hat{t}) \bullet \mathbf{p}(\hat{t})} \text{ is minimal if } f(\hat{t}) = \mathbf{p}(\hat{t}) \bullet \mathbf{p}(\hat{t}) \text{ is minmal}$$

$$f(\hat{t}) = \left(p_0 + \hat{t}\mathbf{v} \right) \bullet \left(p_0 + \hat{t}\mathbf{v} \right) = p_0 \bullet p_0 + 2\hat{t} p_0 \bullet v + \hat{t}^2 \mathbf{v} \bullet \mathbf{v}$$

$$\frac{df}{d\hat{t}} = 2 p_0 \bullet v + 2\hat{t}\mathbf{v} \bullet \mathbf{v} = 0 \Leftrightarrow \hat{t} = \frac{-p_0 \bullet v}{\mathbf{v} \bullet \mathbf{v}}$$

$$d = \left|\mathbf{p}(\hat{t})\right| = \left| p_0 - \frac{p_0 \bullet \mathbf{v}}{\mathbf{v} \bullet \mathbf{v}} \mathbf{v} \right|$$

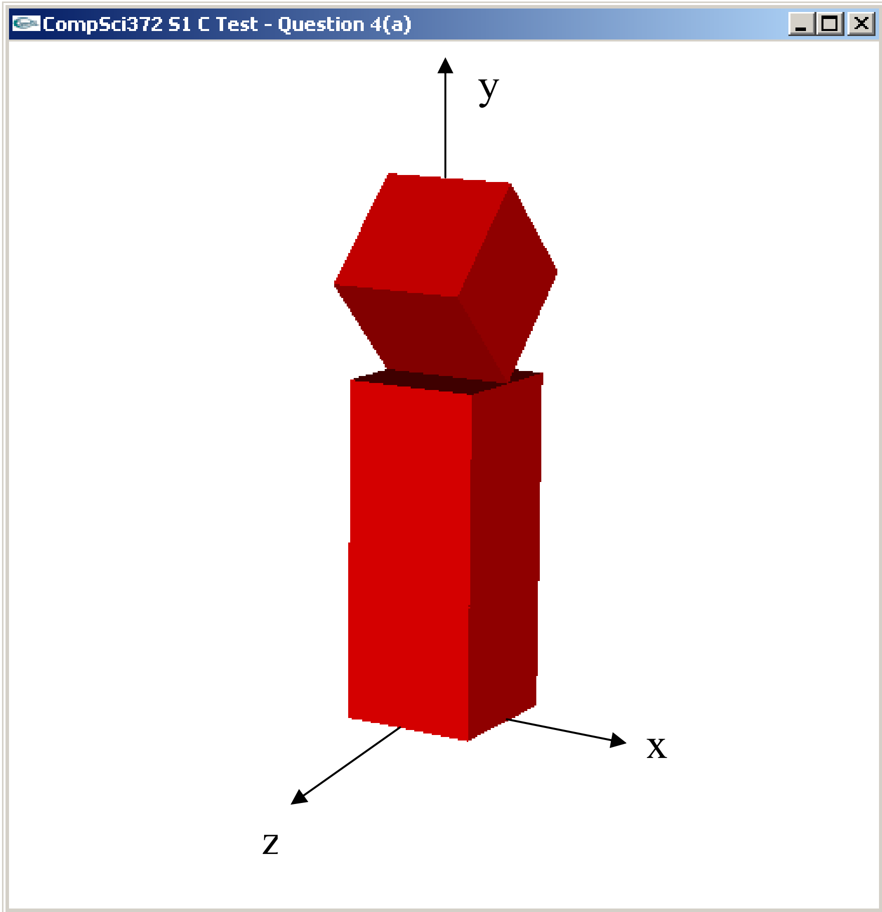
Solution 4: (by comparing Triangle areas) $\quad d = \dfrac{\left|(\mathbf{v} + \mathbf{p}_0) \times \mathbf{p}_0\right|}{|\mathbf{v}|}$

Solution 5: (projection of $\mathbf{p}_0$ onto a vector n which is orthogonal to the line and pointing to the origin) $\quad \mathbf{n} = \left(\mathbf{p}_0 \times \mathbf{v} \right) \times \mathbf{v} \quad d = \dfrac{\left|\mathbf{p}_0 \bullet \mathbf{n}\right|}{|\mathbf{n}|}$

Solution 6: (minimizing |p(t)|^2 algebraically) $\quad d = \sqrt{\left(\mathbf{p}_0 \bullet \mathbf{p}_0\right) - \dfrac{\left(\mathbf{v} \bullet \mathbf{p}_0\right)^2}{\left(\mathbf{v} \bullet \mathbf{v}\right)}}$

Question 4 – Transformations [12 marks]

A. Given is a function *drawCube()* which draws an axis-aligned **unit cube** with side length 1 **centred** at the origin.
Use this function to complete the display method on the next page so that it draws the i-shaped object shown in the image below. The bottom part of the object has a height of two units and a unit square base centred at the origin. The top part is a unit cube rotated by 45 degrees. [6 marks].

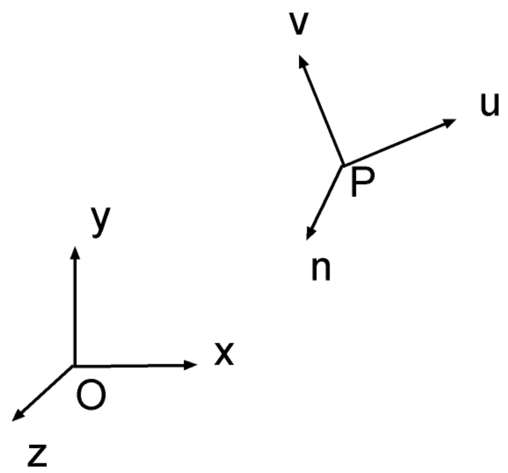


```
#include <math.h>

void display(void)
{
    glMatrixMode( GL_MODELVIEW ); // Set the view matrix ...
    glLoadIdentity();              // ... to identity.
    gluLookAt(0,0,8, 0,1.2,0, 0,1,0);  // camera is on the z-axis
    trackball.tbMatrix();          // rotate the object using the trackball ...
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    // set material properties
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mat_ambient_and_diffuse);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);

    // Draw the scene
```

```
    glPushMatrix();
    glScalef(1,2,1);              // scale in y-direction
    glTranslatef(0,0.5f,0);       // first translate so that base of cube on ground plane
    drawCube();
    glPopMatrix();
    glTranslatef(0,2+sqrt(2)/2,0); // length of bottom shape plus
                                   // half the length of diagonal of cube
    glRotatef(45,1,0,0);          // rotate unit cube
    drawCube();
```

```
    glFlush ();
    glutSwapBuffers();
}
```

B. Given is a world coordinate system with the origin O=(0,0,0) and a second coordinate system with the origin P=(p_x, p_y, p_z) in world coordinates. The basis vectors **u**, **v** and **n** of the second coordinate system are normalised and orthogonal to each other.
Find the homogenous matrix **M** which converts the *uvn*-coordinate system into the world coordinate system. Write down all components of the matrix **M**. [6 marks]



1. Translate origin of uvn-system into origin of the world-coordinate system
2. Rotate uvn-axes into xyz-axes.

$$\mathbf{M} = \mathbf{R}_{uvn}^{-1}\mathbf{T}_{-p} = \begin{pmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} u_x & u_y & u_z & -\mathbf{p}\bullet\mathbf{u} \\ v_x & v_y & v_z & -\mathbf{p}\bullet\mathbf{v} \\ n_x & n_y & n_z & -\mathbf{p}\bullet\mathbf{n} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Question 5 – Modelling and texture mapping [6 marks]

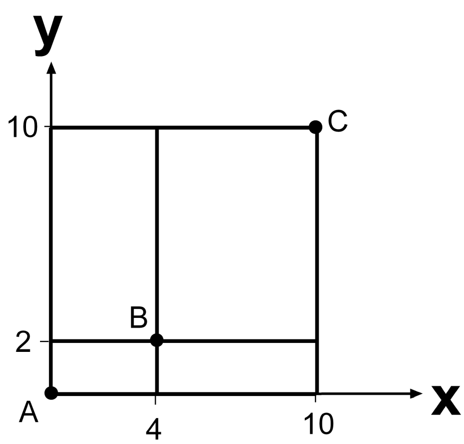
A. Determine the parametric equation of the surface of revolution $\mathbf{p}(s,t)=(x(s,t),y(s,t),z(s,t))$ formed by rotating a line segment $\mathbf{c}(t)=(x(t),y(t),0)$ around the x-axis [2 mark].

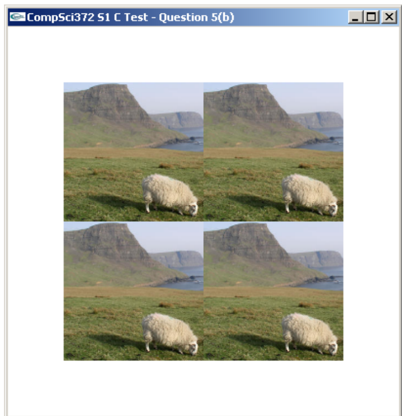
$$p(s,t)=\begin{pmatrix} x(t) \\ y(t)\cos s \\ y(t)\sin s \end{pmatrix} \quad \text{where } s\in[0,2\pi]$$

B. Given is the texture shown in the image below on the left and four polygons shown in the image below in the middle. Assume the following texture parameters are set:

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
```

If the vertex A has the texture coordinates (0,0) which texture coordinates do you have to define for the vertices B and C in order to get the image shown below on the right? [4 marks].





Texture coordinate for C is (2, 2)
Texture coordinate for A is (0.8, 0.4)