단국대학교 멀티미디어공학 그래픽스 프로그래밍 중간고사 (2007년 봄학기) 2007년 4월 26일

학과 _____ 학번 _____ 이름 _____

중간고사

담당교수: 단국대학교 멀티미디어공학전공 박경신

- **답은 반드시 답안지에 기술할 것. 공간이 부족할 경우 반드시 답안지 몇 쪽의 뒤에 있다고 명기한 후 기술할 것. 그 외의 경우의 답안지 뒤쪽이나 연습지에 기술한 내용은 답안으로 인정 안 함. 답에는 반드시 네모를 쳐서 확실히 표시할 것.**
- **답안지에 학과, 학번, 이름 외에 본인의 암호를 기입하면 성적공고시 학번 대신 암호를 사용할 것임.**

1. 맞으면 true, 틀리면 false를 적으시오. (20점)

1) 행렬 (**A B**) **C** = **A** (**B C**) _____T___
2) 벡터 $a \cdot b > 0$ 이면 두 벡터 간의 각도는 $90^\circ \le \theta < 180^\circ$ 이다. ____F___
3) glColor3f(1.0, 1.0, 0.0)은 보라색을 나타낸다. ____F____
4) 하나의 윈도우 안에 싱글버퍼와 더블버퍼를 동시에 사용할 수 없다. ____T___
5) 비트맵(bitmap)의 문자는 크기가 변하지 않는다. ____T___
6) 볼록한(convex) 객체는 객체 내에 임의의 두 점을 연결하는 선분 위에 놓인 임의의 점이 객체 내에 있다. ____T___
7) 더블 버퍼링(double buffering)을 사용하면 프로그램의 frame rate이 증가된다. ___F___
8) 오른손 좌표계 (right-handed coordinate system)에서는 반시계 방향 (counter-clockwise) 회전 (rotation)일 때 θ가 양수이다. ___T____
9) 크기변환 (scaling)은 아핀강체변환(affine rigid-body transformation)이다. ____F___
10) 하나의 점과 두 개의 평행하지 않는 벡터에 의해 평면(plane)이 결정된다. ___T____

2. 다음 문제에 답하시오. (50점)

1) RGB 색깔 모델로 표현된 색깔 (0.5, 0.2, 0.7)을 CMY 색깔 모델을 사용하여 표현했을 때의 값은 무엇인가?

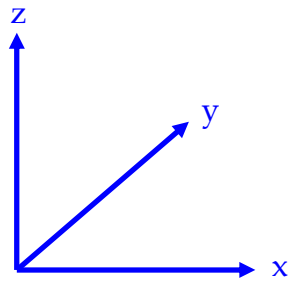
CMY = (1, 1, 1) - RGB
= (1, 1, 1) – (0.5, 0.2, 0.7)
= (0.5, 0.8, 0.3)

2) 더블 버퍼링 (Double buffering)에 대해 설명하시오.

비디오 제어기가 항상 완성된 이미지를 도시하도록 함.
이를 위하여 프로세서는 이미지의 내용을 계산하여 후면 버퍼에 축적.
그동안 비디어 제어기는 전면 버퍼의 내용을 읽어 화면에 이미지 도시함.
싱글 버퍼링보다 훨씬 부드러운 애니메이션 생성.
더블 버퍼링을 사용할 경우 버퍼가 하나 더 필요함. 필요에 따라 주어진 색깔 버퍼를 두 개로 나누어야 함 (전면 버퍼와 후면 버퍼).
실제 frame rate이 낮아질 수 있음.

학과 _____ 학번 _____ 이름 _____

3) +x축이 오른쪽, +y축이 화면의 안쪽으로 들어가는 방향일 때, 오른손 좌표계 (right-handed coordinate system)를 사용하는 좌표계에서 +z축이 가리키는 방향이 어느 곳인지 x, y, z축을 그리시오.



4) 동차좌표(homogeneous coordinates)로 표현된 3차원 공간의 점 (1, 0, -3, 2)과 동일한 아핀공간 (affine space)에서의 점의 좌표 (x, y, z)는 무엇인가?

(1, 0, -3, 2) -> (1/2, 0, -3/2)

5) (1, 2, 3)만큼 이동(translate)한 후, z축으로 30도 회전(rotate)한 4x4 행렬 (matrix)를 계산하시오. (cos30°, sin30°은 그대로 사용할 것)

$$\begin{pmatrix} \cos30 & -\sin30 & 0 & 0 \\ \sin30 & \cos30 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

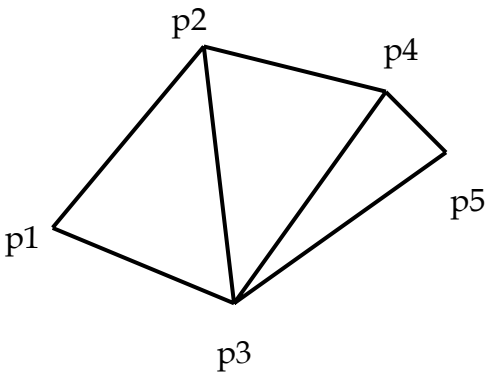
$$= \begin{pmatrix} \cos30 & -\sin30 & 0 & \cos30 - 2\sin30 \\ \sin30 & \cos30 & 0 & \sin30 + 2\cos30 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

6) 다음 OpenGL 함수 중 기본 요소가 화면 상에 나타날 수 있는 방법을 제어하는 속성 (attributes) 함수를 모두 동그라미 표시 하시오.

glBegin	glEnd	glTranslate
glColor3f	gluOrtho2D	glPointSize
glVertex3f	glPolygonMode	glLineWidth

학과 _____ 학번 _____ 이름 _____

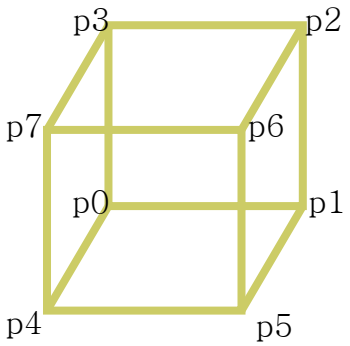
7) OpenGL의 glBegin(GL_TRIANGLES), glEnd(), glVertex를 사용하여 아래와 같은 그림을 그리는 간단한 OpenGL code를 작성하시오.




```
glBegin(GL_TRIANGLES);
    glVertex2iv(p1);
    glVertex2iv(p2);
    glVertex2iv(p3);
    glVertex2iv(p2);
    glVertex2iv(p3);
    glVertex2iv(p4);
    glVertex2iv(p3);
    glVertex2iv(p4);
    glVertex2iv(p5);
glEnd();
```

8) 다음은 입방체(cube)를 그리는 OpenGL 프로그램의 일부이다. 빈칸을 채우시오.

```
void drawCube()
{
    glBegin(GL_QUADS);
    …

    glNormal3f( -1.0 ,   0.0 ,   0.0 ); // left
    glVertex3fv( p4 );
    glVertex3fv( p7 );
    glVertex3fv( p3 );
    glVertex3fv( p0 );

    glNormal3f(0.0, -1.0, 0.0); // bottom
    glVertex3fv(p4);
    glVertex3fv(p0);
    glVertex3fv(p1);
    glVertex3fv(p5);
    ...

    glEnd();
}
```

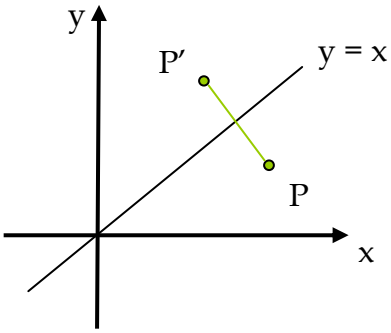


학과 _____ 학번 _____ 이름 _____

9) 투영 변환 (projection transformation)과 관련 있는 OpenGL 함수 2개를 적어라.

glOrtho
gluOrtho2D
glFrustum
gluPerspective

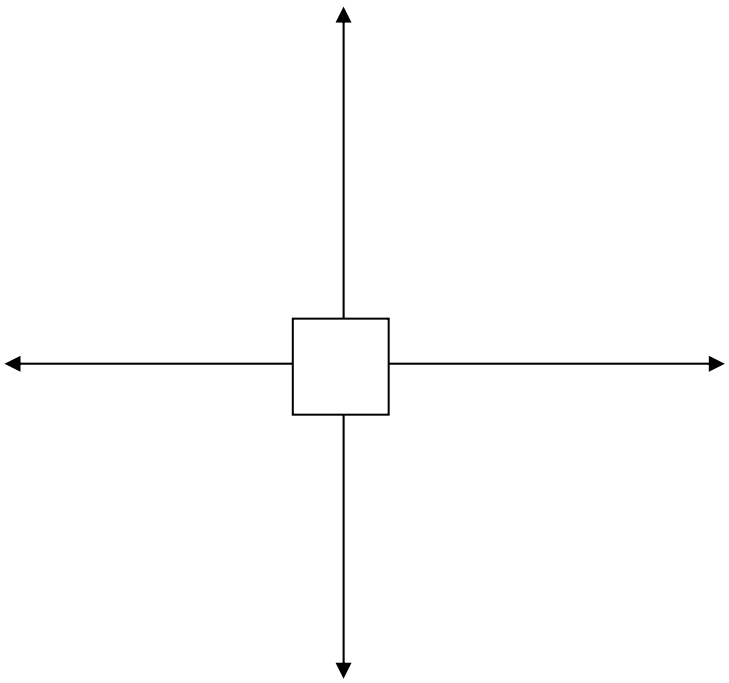
10) 아래 그림에서와 같이 2차원 공간의 점 P(x, y)를 직선 L (y = x)에 대하여 반사 (reflection)시켜 P′(x′, y′)으로 변환시키는 아핀 변환을 유도하라.



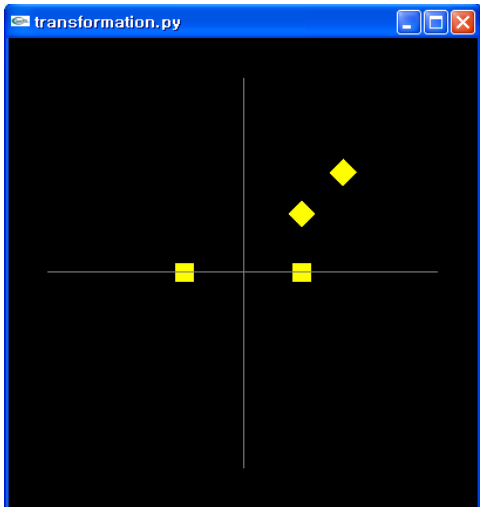
x′ = y
y′ = x

3. drawSquare()는 (0,0) 원점에서 1 x 1 정사각형을 그려주는 함수이다. 아래에 주어진 간단한 코드를 이해하고 World Coordinate System을 중심으로 하여 그림을 그려라. (10점)

```
glLoadIdentity();
glTranslatef(3, 0, 0);
drawSquare();
glPushMatrix();
glTranslatef(-6, 0, 0);
drawSquare();
glPopMatrix();
glPushMatrix();
glTranslatef(0, 3, 0);
glRotatef(45, 0, 0, 1);
drawSquare();
glPushMatrix();
glTranslatef(3, 0, 0);
drawSquare();
glPopMatrix();
glPopMatrix();
```



학과 _____ 학번 _____ 이름 _____



4. 임의의 회전축 a [a_x, a_y, a_z] (a는 단위벡터)에 대해 θ만큼 회전 (rotate)하는 행렬 (matrix)는 아래의 공식과 같이 간단히 정의될 수 있다. 이 때, Symmetric matrix A는 A와 v (x, y, z)를 곱한 것 *A*v = a(a · v)이다. 또한, Skew matrix B는 B와 v (x, y, z)를 곱한 것 *B*v = a x v이다.
이 공식을 이용하여 4x4 행렬 (matrix) R을 구하라. (10점)
R = ***I*** cosθ + ***Symmetric*** (1 – cosθ) + ***Skew*** sinθ

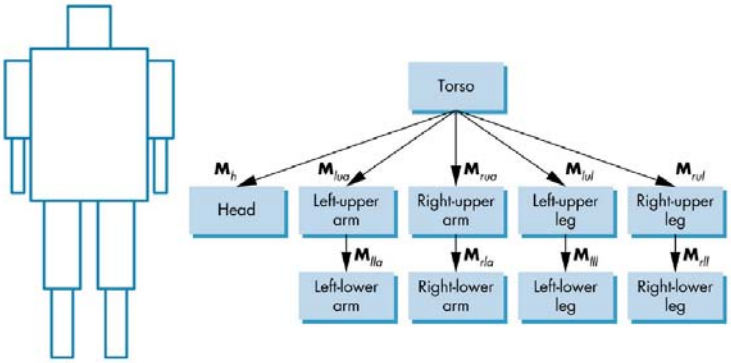
$$R = \mathrm{I}\cos\theta + \textbf{Symmetric}\quad (1-\cos\theta) + \textbf{Skew}\ \sin\theta$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\cos\theta + \begin{bmatrix} a_x^2 & a_x a_y & a_x a_z \\ a_x a_y & a_y^2 & a_y a_z \\ a_x a_z & a_y a_z & a_z^2 \end{bmatrix}(1-\cos\theta) + \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}\sin\theta$$

$$= \begin{bmatrix} a_x^2(1-\cos\theta)+\cos\theta & a_x a_y(1-\cos\theta)-a_z\sin\theta & a_x a_z(1-\cos\theta)+a_y\sin\theta \\ a_x a_y(1-\cos\theta)+a_z\sin\theta & a_y^2(1-\cos\theta)+\cos\theta & a_y a_z(1-\cos\theta)-a_x\sin\theta \\ a_x a_z(1-\cos\theta)-a_y\sin\theta & a_y a_z(1-\cos\theta)+a_x\sin\theta & a_z^2(1-\cos\theta)+\cos\theta \end{bmatrix}$$

학과 _____ 학번 _____ 이름 _____

5. 다음은 로봇의 계층적 변환 (hierarchical transformation)을 표현한 구조를 보여주고 있다. glPushMatrix(), glPopMatrix(), glTranslatef(), glRotatef()을 사용하여 로봇을 표현하는 display()함수를 만들어라.




```
void display()
{
    glPushMatrix() ;
    torso() ;
        glPushMatrix() ;
        glTranslatef() ;
        glRotatef() ;
        head() ;
        glPopMatrix() ;
        glPushMatrix() ;
        glTranslatef() ; glRotatef() ;
        Left_upper_arm() ;
            glTranslatef() ; glRotatef() ;
            Left_lower_arm() ;
        glPopMatrix() ;
        glPushMatrix() ;
        glTranslatef() ; glRotatef() ;
        Right_upper_arm() ;
            glTranslatef() ; glRotatef() ;
            Right_lower_arm() ;
        glPopMatrix() ;
        glPushMatrix() ;
        glTranslatef() ; glRotatef() ;
        Left_upper_leg() ;
            glTranslatef() ; glRotatef() ;
            Left_upper_leg() ;
```

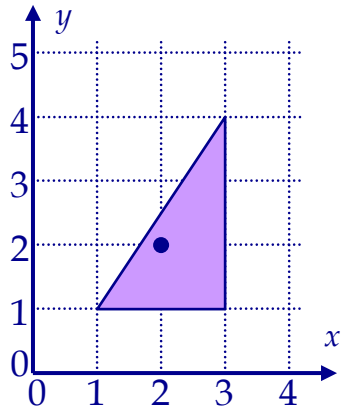
```
            glPopMatrix() ;
            glPushMatrix() ;
            glTranslatef() ; glRotatef() ;
            Right_upper_leg() ;
                  glTranslatef() ; glRotatef() ;
                  Right_lower_leg() ;
            glPopMatrix() ;
      glPopMatrix() ;
}
```

6. 다음은 vector class의 일부를 보여주고 있다. 두 벡터 간의 내적 (dot product)과 두 점 간의 거리(distance)를 구하는 함수를 작성하시오. (보너스 문제 extra 10점)

```
class vector
{
public:
        float x, y, z;
        vector(void) { x = y = z= 0.0f; }
        vector(float x_, float y_, float z_) { x = x_; y = y_; z = z_; }
        ...

        static float dot(const vector &v1, const vector &v2);
        static float distance(const vector &v1, const vector & v2);
};

static float vector::dot (const vector &v1, const vector &v2)
{
        return (v1.x * v2.x + v1.y * v2.y + v1.z * v2.z);
}

static float vector::distance (const vector &v1, const vector &v2)
{
        float dx = v1.x - v2.x;
        float dy = v1.y - v2.y;
        float dz = v1.z - v2.z;
        return (float)sqrt( dx * dx + dy * dy + dz * dz );
}
```

학과 _____ 학번 _____ 이름 _____

7. 정점 (1, 1), (3, 1), (3, 4)를 갖는 삼각형을 점 (2, 2)에 대하여 x-축으로 -1만큼, y-축으로 2만큼 크기 변환하라. 먼저 변환 행렬 (matrix)를 계산하고 변환된 각 정점의 값을 계산하시오. (보너스 문제 extra 10점) 답 **(3,0), (1, 0), (1, 6)**



$$M = T_{(2,2)}S(-1,2)T_{(-2,-2)}$$

$$= \left(\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \right)$$

$$= \begin{bmatrix} -1 & 0 & 2 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 0 & 4 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{M}$$

$$P_1' = MP_1 = \begin{bmatrix} -1 & 0 & 4 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}$$

$$P_2' = MP_2 = \begin{bmatrix} -1 & 0 & 4 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$P_3' = MP_3 = \begin{bmatrix} -1 & 0 & 4 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ 1 \end{bmatrix}$$