

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

2. 다음 문제에 답하시오. (50점)

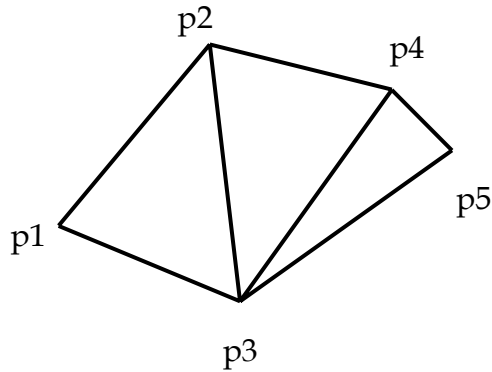
- 1) RGB 색깔 모델로 표현된 색깔 (0.5, 0.2, 0.7)을 CMY 색깔 모델을 사용하여 표현했을 때의 값은 무엇인가?

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

- 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?
- 5) (1, 2, 3)만큼 이동(translate)한 후, z축으로 30도 회전(rotate)한 4x4 행렬 (matrix)를 계산하시오. ($\cos 30^\circ$, $\sin 30^\circ$ 은 그대로 사용할 것)
- 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

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



```
glBegin(GL_TRIANGLES);
```

```
glEnd();
```

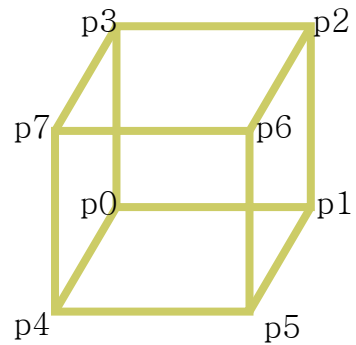
- 8) 다음은 입방체(cube)를 그리는 OpenGL 프로그램의 일부이다. 빈칸을 채우시오.

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

    glNormal3f(____, ____ , ____); // left
    glVertex3fv(____);
    glVertex3fv(____);
    glVertex3fv(____);
    glVertex3fv(____);

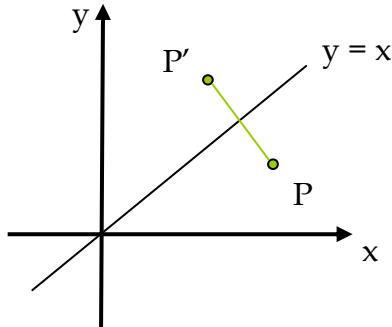
    glNormal3f(____, ____ , ____); // bottom
    glVertex3fv(____);
    glVertex3fv(____);
    glVertex3fv(____);
    glVertex3fv(____);
    ...

    glEnd();
}
```



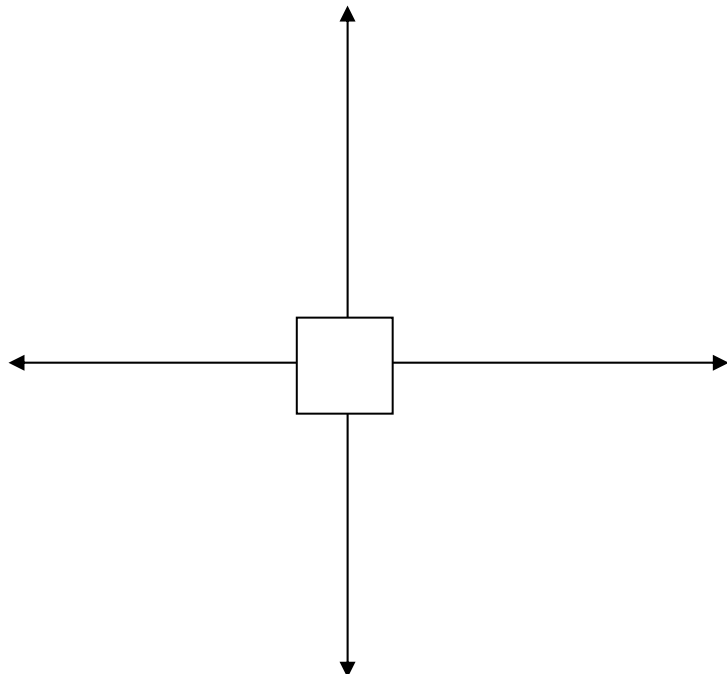
9) 투영 변환 (projection transformation)과 관련 있는 OpenGL 함수 2개를 적어라.

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.



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();
```

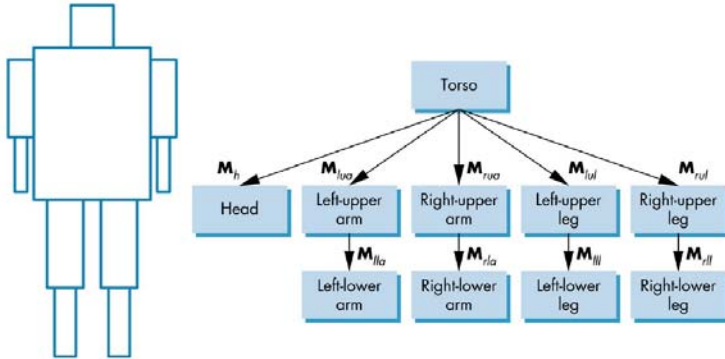


4. 임의의 회전축 $\mathbf{a} [a_x, a_y, a_z]$ (\mathbf{a} 는 단위벡터)에 대해 θ 만큼 회전 (rotate)하는 행렬 (matrix)는 아래의 공식과 같이 간단히 정의될 수 있다. 이 때, Symmetric matrix \mathbf{A} 는 \mathbf{A} 와 $\mathbf{v} (x, y, z)$ 를 곱한 것 $\mathbf{A}\mathbf{v} = \mathbf{a}(\mathbf{a} \cdot \mathbf{v})$ 이다. 또한, Skew matrix \mathbf{B} 는 \mathbf{B} 와 $\mathbf{v} (x, y, z)$ 를 곱한 것 $\mathbf{B}\mathbf{v} = \mathbf{a} \times \mathbf{v}$ 이다. 이 공식을 이용하여 4x4 행렬 (matrix) \mathbf{R} 을 구하라. (10점)

$$\mathbf{R} = \mathbf{I} \cos\theta + \text{Symmetric} (1 - \cos\theta) + \text{Skew} \sin\theta$$

$$\mathbf{R} = \left(\begin{array}{ccc} & & \\ & & \\ & & \end{array} \right)$$

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();
```

```
    glPopMatrix();
```

```
}
```

6. 다음은 `vector class`의 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_; }
    ...
    float length(void);
    void normalize(void);
    static float dot(const vector &v1, const vector &v2);
    static float distance(const vector &v1, const vector &v2);
};
```

```
float vector::length( void )
```

```
{
    return( (float)sqrt( x * x + y * y + z * z ) );
}
```

```
void vector::normalize( void )
```

```
{
    float fLength = length();
    x = x / fLength;
    y = y / fLength;
    z = z / fLength;
}
```

```
static float vector::dot (const vector &v1, const vector &v2)
```

```
{
```

```
}
```

```
static float vector::distance (const vector &v1, const vector &v2)
```

```
{
```

```
}
```

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

