학과 _____ 학번 _____ 이름 _____

중간고사

담당교수: 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. 다음 문제에 답하시오. (총 25점)

1) **깊이 버퍼(Depth buffer)가 무엇인지? OpenGL/GLUT에서 깊이 버퍼를 사용하기 위해 필요로 하는 설정을 서술하라. (5점)**

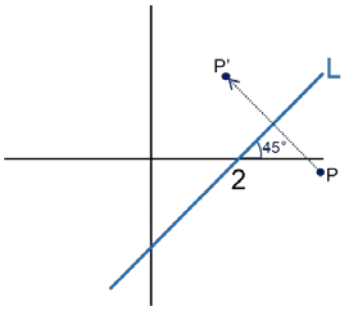
2) **더블 버퍼링 (Double buffering)이 무엇인지? OpenGL/GLUT에서 더블 버퍼링을 사용하기 위해서 필요로 하는 설정을 서술하라. (5점)**

3) **OpenGL에서 기본 카메라(default camera)에 대하여 간단히 서술하라. (5점)**

4) **glutDisplayFunc()와 glutRedisplay() 함수가 무엇인지 간단히 서술하라. (5점)**

5) **Modern OpenGL 렌더링을 하기 위해 사용하는 Vertex Array Object (VAO), Vertex Buffer Object (VBO), Index Buffer Object (IBO)이 무엇인지 간단히 설명하라. (5점)**

2. 다음은 2차원 공간의 점 P(x, y)를 직선 L에 대하여 반사(reflection)된 점 P'(x', y')에 대한 2차원 아핀 변환 행렬 (3x3 matrix)에 관한 문제이다. 아래의 물음에 답하라. (각 5점 총 25점)



1) 2차원 공간에 임의의 두 점을 선택하여 직선 L에 대한 반사된 점들을 계산하라.

2) 2차원 공간에 임의의 점 P을 직선 L에 대하여 반사해서 점 P'으로 변환시키는 아핀 변환 행렬 M (3x3)을 유도하라.

$$M = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

3) 위(2.2)의 아핀 변환 행렬 M을 GLM translate(이동), rotate(회전), scale(크기) 등의 기본 변환 행렬들의 곱으로 적절히 표현하라. 행렬의 곱의 순서에 주의할 것. 아래는 GLM translate, rotate, scale 변환 함수의 사용법을 보여준다.

```
glm::mat4 M1 = glm::translate(glm::mat4(1), glm::vec3(dx, dy, dz));
glm::mat4 M2 = glm::rotate(glm::mat4(1), glm::radian(angle), glm::vec3(ax, ay, az));
glm::mat4 M3 = glm::scale(glm::mat4(1), glm::vec3(sx, sy, sz));
```

4) 위(2.3)의 행렬을 직접 행렬의 곱을 계산하여 (2.2)와 같음을 증명한다. 3x3 아핀 변환 행렬을 사용할 것.

$M =$

5) 임의의 두 점(2.1)을 이용하여 직선 L에 반사되는 행렬 M(2.2)을 곱하여 변환된 점들이 (2.1)과 같은 값이 나옴을 증명한다.

$P_1' = M * P_1 =$

$P_2' = M * P_2 =$

3. 다음은 3차원 공간에서 임의의 축 (arbitrary axis) (1, 1, 0)에 대한 45도 각도 회전 Rotation Vector를 계산하는 과정이다. 아래의 질문에 답하라. (각 3점 총 18점)

1) 임의의 축 a(1, 1, 0)를 정규화한 $\bar{a}$을 구하라.

$\bar{a} =$

2) $\bar{a}$와 정점 X(3, 3, -1)간의 외적(cross product) w를 계산하라.

$$W = \bar{a} \times X =$$

3) 정점 X(3, 3, -1)이 $\bar{a}$에 평행한 벡터 $X_{\parallel}$를 계산하라.

$$X_{||} =$$

4) 정점 X(3, 3, -1)이 $\bar{a}$에 수직인 벡터 $X_{\perp}$를 계산하라.

$$X_{\perp} =$$

5) 벡터 $X_{\perp}$가 $\bar{a}$에 45도 회전한 벡터 $R(X_{\perp})$를 계산하라.

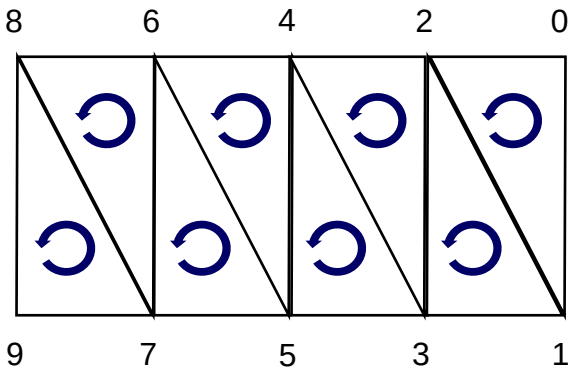
$$R(x_{\perp}) =$$

6) 정점 X(3, 3, -1)이 임의의 축 a 에 45 회전한 벡터 R(X)를 계산하라.

$$R(x) =$$

학과 _____ 학번 _____ 이름 _____

4. GL_LINES, GL_LINE_STRIP, GL_TRIANGLES, GL_TRIANGLE_STRIP을 사용하여, 다음 도형을 wireframe/fill 그리기 위한 정점 리스트(정점 인덱스 사용)를 적어라. (각 3점 총 12점)



GL_LINES	GL_LINE_STRIP	GL_TRIANGLES	GL_TRIANGLE_STRIP

학과 _____ 학번 _____ 이름 _____

5. 다음 물음에 답하시오. (20점)

1) SimpleCar 클래스의 계층적 변환(hierachical transformation) 구조를 3차원 공간에서 정확한 척도로 표시하라. 또한 이 계층적 변환 구조에서 중심의 위치를 표시해준다. (10점)

```
// SimpleCar.cpp
void SimpleCar::init()
{
	geo[0] = Parallelepiped(glm::vec3(0, 0, 0), glm::vec3(1, 0, 0), glm::vec3(0, 0, 1), glm::vec3(0, 1, 0));
	geo[0].setColor(glm::vec3(0, 1, 0)); // green parallelepiped
	geo[1] = Torus(glm::vec3(0, 0, 0), 0.5, 0.1, 16, 16);
	geo[1].setColor(glm::vec3(1, 1, 0)); // yellow torus
	geo[2] = Torus(glm::vec3(0, 0, 0), 0.5, 0.1, 16, 16);
	geo[2].setColor(glm::vec3(0, 0, 0)); // black torus
}

void SimpleCar::draw(Program* p, glm::mat4& projection, glm::mat4& view, glm::mat4& model)
{
	p->useProgram();
	p->setUniform("gProjection", projection);
	p->setUniform("gView", view);

	mat4 mMatrix = model * scale(mat4(1), vec3(4, 2, 4));
	p->setUniform("gModel", mMatrix);
	geo[0].draw(); // (1) lower body
	mMatrix = model * translate(mat4(1), vec3(1, 2, 1)) * scale(mat4(1), vec3(2, 1, 2));
	p->setUniform("gModel", mMatrix);
	geo[0]->draw(); // (2) upper body

	mMatrix = model * translate(mat4(1), vec3(4, 1, 1)) * rotate(mat4(1), M_PI/2, vec3(0, 1, 0)) *
scale(mat4(1), vec3(0.5, 0.5, 2));
	p->setUniform("gModel", mMatrix);
	geo[1]->draw(); // (3) left head light
	mMatrix = model * translate(mat4(1), vec3(4, 1, 3)) * rotate(mat4(1), M_PI/2, vec3(0, 1, 0)) *
scale(mat4(1), vec3(0.5, 0.5, 2));
	p->setUniform("gModel", mMatrix);
	geo[1]->draw(); // (4) right head light

	mMatrix = model * translate(mat4(1), vec3(1, 0, 0));
	p->setUniform("gModel", mMatrix);
	geo[2]->draw(); // (5) left back wheel
	mMatrix = model * translate(mat4(1), vec3(1, 0, 4));
	p->setUniform("gModel", mMatrix);
	geo[2]->draw(); // (6) right back wheel
	mMatrix = model * translate(mat4(1), vec3(3, 0, 0));
	p->setUniform("gModel", mMatrix);
	geo[2]->draw(); // (7) left front wheel
	mMatrix = model * translate(mat4(1), vec3(3, 0, 4));
	p->setUniform("gModel", mMatrix);
	geo[2]->draw(); // (8) right front wheel
}
```

2) main.cpp 의 draw()함수의 실행 결과를 보여라. 3차원 공간에서 정확한 척도로 표시한다. 또한 car의 중심 위치를 표시해준다. (10점)

```
// main.cpp
void draw()
{
	glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );

	// MVP matrix
	Projection = glm::perspective(g_fovy, g_aspect, g_zNear, g_zFar);
	View = glm::lookAt(g_eye, g_at, g_up);
	spMain.useProgram();
	spMain.setUniform("gProjection", Projection);
	spMain.setUniform("gView", View);

	glm::mat4 R = glm::rotate(glm::mat4(1.0f), glm::radians(-90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
	World = R;
	spMain.setUniform("gModel", World);
	car->draw(&spMain, Projection, View, World); //(1)

	glm::mat4 T2 = glm::translate(glm::mat4(1.0f), glm::vec3(5.0f, 0.0f, 0.0f));
	World = T2 * R;
	car->draw(&spMain, Projection, View, World); //(2)
}
```

- 끝 -