

2010학년도 컴퓨터그래픽스 기말시험 기출문제 해설

36. 다음 중 3차원 데카르트 좌표 (2, 3, 4)에 해당되는 동차좌표는 무엇인가? (3점)

- ① (2, 3, 4, 0) ② (2, 3, 4, 5)
 ③ (4, 6, 8, 2) ④ (3, 4, 5, 2)

정답 : ③

해설 : 3차원 데카르트 좌표 (x, y, z) 를 동차좌표 $(x \cdot h, y \cdot h, z \cdot h, h)$ 로 표현할 수 있다. 이때 h 는 동차 파라미터이며, x, y, z 좌표에 동차 파라미터를 곱하면 동차 좌표를 구할 수 있다. ③의 경우 동차 파라미터가 2이며, 문제의 x, y, z 좌표 2, 3, 4에 각각 2를 곱하여 구한 동차좌표이다.

교재 관련 부분 : 154쪽

37. 다음 중 x 축을 중심축으로 하여 시계반대 방향으로 ϕ 만큼 회전하는 변환에 대한 행렬은? (3점)

- ① $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ ② $\begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
 ③ $\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & \cos\phi & -\sin\phi & 1 \\ 0 & \sin\phi & \cos\phi & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ ④ $\begin{bmatrix} \cos\phi & -\sin\phi & 0 & 1 \\ \sin\phi & \cos\phi & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

정답 : ①

해설 : 변환 수식을 이해한다면 다음의 변환행렬을 쉽게 유도할 수 있다.

변환		변환행렬
x, y, z 축 방향으로 각각 t_x, t_y, t_z 만큼 이동		$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$
x, y, z 축 방향으로 각각 s_x, s_y, s_z 배 만큼 크기조정		$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
회전	x 축 중심으로 반시계방향 ϕ 만큼 회전	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

	y축 중심으로 반시계방향 ϕ 만큼 회전	$\begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	z축 중심으로 반시계방향 ϕ 만큼 회전	$\begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

교재 관련 부분 : 156~161쪽

38. 점(3, 4)를 기준으로 하여 x 축 방향으로 2배, y 축 방향으로 3배 확대하는 크기변환은? ($T(t_x, t_y)$ 는 이동, $S(s_x, s_y)$ 는 크기변환을 나타낸다.) (3점)

- ① $S(2, 3) \cdot T(-3, -4)$
- ② $T(3, 4) \cdot S(2, 3) \cdot T(3, 4)$
- ③ $S(2, 3) \cdot T(3, 4)$
- ④ $T(3, 4) \cdot S(2, 3) \cdot T(-3, -4)$

정답 : ④

해설 : 점 P 에 변환행렬 M_1 을 가하여 변환된 좌표는 $P_1 = M_1 \cdot P$ 이고, 이에 다시 변환행렬 M_2 를 가하여 변환된 좌표는 $P_2 = M_2 \cdot P_1 = M_2 \cdot M_1 \cdot P$ 이다. 그러므로 이 복합변환의 행렬은 $M_2 \cdot M_1$ 이다. 즉, 먼저 해야 할 변환행렬을 우측에, 나중에 할 변환행렬을 좌측에 배치하여 곱해 복합변환행렬을 구한다. 점(3, 4)를 기준으로 하여 x 축 방향으로 2배, y 축 방향으로 3배 확대하는 것은 (1) 좌표를 x 축 방향으로 -3, y 축 방향으로 -4만큼 이동($T(-3, -4)$)하여 (3, 4)가 원점으로 이동하도록 한 후 (2) 원점을 기준으로 하여 x 축 방향으로 2배, y 축 방향으로 3배 확대($S(2, 3)$)한 다음 (3) 다시 원점이 (3, 4)로 이동하도록 x 축 방향으로 3, y 축 방향으로 4만큼 이동($T(3, 4)$)하는 것이다. 그러므로 순서에 유의하여 각각의 변환행렬을 곱하면 문제의 복합변환행렬은 $T(3, 4) \cdot S(2, 3) \cdot T(-3, -4)$ 가 된다.

교재 관련 부분 : 162, 164~165쪽

39. 강체변환에 대한 올바른 설명은? (4점)

- ① 크기변환이 포함된다.
- ② 모든 아핀변환은 강체변환이다.
- ③ 이동과 회전변환은 강체변환이다.
- ④ 밀림변환은 강체변환에 해당된다.

정답 : ③

해설 : 물체의 크기나 모양이 변하지 않는 변환으로서, 회전이나 이동, 이들의 복합 변환인 경우에 강체변환이 된다. 아핀변환은 회전, 이동, 크기, 밀림 등의 변환이 모두 포함되므로 모든 아핀변환이 강체변환은 아니다.

정답 : ③

해설 : Sutherland-Hodgeman 다각형 클리핑은 좌측 경계에 대해 클리핑을 하고, 그 결과를 다시 우측 경계에 대해, 그 다음은 하단 경계, 그 다음은 상단 경계에 대해 클리핑을 수행하는, 즉, 좌, 우, 하, 상 클리핑 경계에 대해 차례로 클리핑을 하는 알고리즘이다. 볼록 다각형의 클리핑 결과는 역시 볼록 다각형이 될 것이다. 그러나 오목 다각형의 경우 때로는 2개의 다각형으로 분리되어야 할 경우가 있는데, Sutherland-Hodgeman 알고리즘을 적용할 경우 변이 겹치는 비 표준 다각형이 나올 수 있다. (교재 212쪽의 그림 6-11)

교재 관련 부분 : 209~211쪽

43. 3차원 뷰잉 파이프라인에서 좌표계들의 변환 순서를 바르게 나열한 것은? (4점)

(㉠) 모델링 좌표	(㉡) 정규 좌표	(㉢) 투영 좌표
(㉣) 세계 좌표	(㉤) 뷰잉 좌표	(㉥) 장치 좌표

- ① (㉠) → (㉡) → (㉢) → (㉣) → (㉤) → (㉥)
- ② (㉠) → (㉣) → (㉤) → (㉢) → (㉡) → (㉥)
- ③ (㉠) → (㉤) → (㉢) → (㉡) → (㉣) → (㉥)
- ④ (㉠) → (㉡) → (㉣) → (㉤) → (㉢) → (㉥)

정답 : ②

해설 : 모델링 좌표계에서 설계된 물체는 적절한 기하변환을 통해 세계좌표에 배치하여 장면을 구성한다. 이 장면을 바라보는 사람의 눈의 위치를 고려하여 뷰잉 좌표계로 변환한 다음, 정해진 투영 방법에 따라 투영 변환을 하여 투영 좌표계에 표현한다. 그 결과에 클리핑 영역을 정의하여 클리핑 및 정규화를 하여 정규 좌표로 변환한 다음 이를 장치 좌표로 변환하여 디스플레이한다.

교재 관련 부분 : 224쪽

44. 다음 중 뷰잉 좌표계의 y_{view} 축의 + 방향을 결정하는 요소는? (3점)

- ① 뷰 평면 법선 벡터
- ② 세계좌표 y 축
- ③ 뷰 지점
- ④ 뷰업 벡터

정답 : ④

해설 : 뷰 지점은 뷰잉 좌표계($x_{view}-y_{view}-z_{view}$ 좌표계)의 원점 위치를 세계좌표계에서 설정하기 위한 파라미터이다. 뷰 평면 법선 벡터는 시야의 깊이 방향을 나타내는 z_{view} 축 방향을 결정한다. $x_{view}-y_{view}$ 축은 뷰 평면의 기준축이다. 이때 x_{view} 은 수평축, y_{view} 축은 수직축에 해당되며, 수직축의 방향을 나타내기 위해 뷰업 벡터를 지정한다. 엄밀하게 뷰업 벡터는 z_{view} 와 수직이어야 할 것이나, $y_{view}-z_{view}$ 평면에 속한 벡터로 적당히 지정하더라도 교재 228쪽의 식(7-1)에 의해 계산할 수 있으며, x_{view} 축 방향은 특별히 지정하지 않더라도 역시 식(7-1)에 의해 계산할 수 있다.

교재 관련 부분 : 225~228쪽

45. 다음 중 뷰 평면이 원점으로부터 동일한 거리에서 세 좌표축과 교차하는 직교투영은? (2점)

- ① 등각투영
- ② 평면도
- ③ 캐버리어 투영
- ④ 캐비닛 투영

정답 : ①

해설 : 직교투영은 투영선이 뷰 평면과 수직인 평행투영으로서, 정면도, 측면도, 평면도는 세계좌표계의 좌표축에 수직인 뷰 평면에 대한 투영이며, 여러 축과 교차하는 뷰 평면에 대한 투영은 측각직교투영이라고 한다. 측각직교투영 중 세 좌표축과 원점으로부터 동일한 거리에서 교차하는 뷰 평면에 대한 투영을 등각투영이라고 한다. 캐버리어 투영과 캐비닛 투영은 투영선과 뷰 평면이 수직이 아닌 경사평행 투영이다.

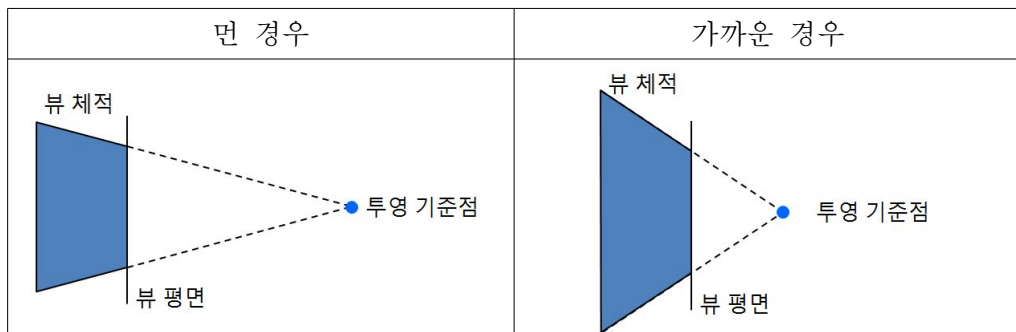
교재 관련 부분 : 230, 231, 236~238쪽

46. 원근투영에 대한 올바른 설명은? (4점)

- ① 원근투영변환은 아핀변환이다.
- ② 투영 기준점이 뷰 평면과 멀어지면 원근 효과가 강조된다.
- ③ 뷰 평면과 평행하지 않은 평행 선분들의 연장선은 한 점으로 모인다.
- ④ 뷰 평면이 $x-y$ 평면과 평행하면 축 소실점은 2개이다.

정답 : ③

해설 : 평행투영은 원근법이 적용된다. 평행한 선들은 무한히 뻗어 가도 실제로는 만나지 않지만, 만일 뷰 평면에 평행하지 않다면 투영 결과는 멀어질수록 점점 가까워져서 결국 한 점에 모이게 되는데, 이를 소실점이라고 한다. 축 소실점(주소실점)은 뷰 평면과 평행하지 않은 주축과 평행한 선들이 모이는 소실점으로서, 뷰 평면이 $x-y$ 평면과 평행하면 뷰 평면과 평행하지 않은 주축은 z 축 하나이며, 따라서 주소실점은 1개이다. ②를 답으로 잘못 선택한 경우가 많은데, 다음 그림을 보면 투영 기준점이 뷰 평면과 가까울수록 원근 효과가 강조됨을 알 수 있다.



교재 관련 부분 : 239~241쪽

n 차 미분이 비례하는 것을 나타낸다. 다음 그림은 p_0 와 p_1 을 잇는 곡선과 p_2 와 p_3 를 잇는 곡선의 기하학적 연결성을 보여준다. 여기에서 p_1' 과 p_1'' 은 p_1 에서 곡선 p_0p_1 의 1차 미분 및 2차 미분을, p_2' 과 p_2'' 은 p_2 에서 곡선 p_2p_3 의 1차 미분 및 2차 미분을 나타낸다.

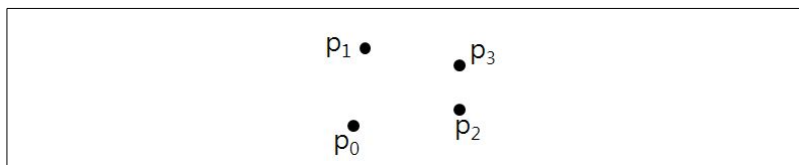
G^0 연속성	G^1 연속성	G^2 연속성

반면 n 차 매개변수 연속성(C^n)은 기하학적 연속성과 유사하나, n 차 미분이 단지 비례하는 것에 머물지 않고 일치하는 것을 의미한다. 다음 그림은 기하학적 연속성과 매개변수 연속성의 차이를 보여준다. 기하학적 연속성에 비해 매개변수 연속성이 더욱 부드러운 곡선 연결을 형성하는 것을 볼 수 있다.

G^1 연속성	C^1 연속성

교재 관련 부분 : 284~286쪽

50. 네 개의 제어점 p_0 , p_1 , p_2 , p_3 가 다음과 같이 차례로 지정되었을 때 생성되는 3차 베지어 곡선의 모양은 무엇인가? (3점)



- ①
- ②
- ③
- ④

정답 : ①

해설 : 베지어 곡선은 제어점들을 둘러싸는 볼록껍질 내에 포함되며, 첫째 제어점과 마

교재 관련 부분 : 290~292쪽

① 2개 ② 3개
③ 4개 ④ 5개

① 허미트 곡선 : C^2 연속성을 갖는다.
 ② 베지어 곡선 : 원근투영 불변성을 갖는다.

- ③ 3차 균일 B-스플라인 : 모든 제어점을 연결한다.
- ④ NURBS : 원뿔곡선을 재현할 수 있다.

정답 : ④

해설 : 허미트 곡선은 G^0 , G^1 , C^1 연속성을 갖는 곡선을 정의할 수 있다. 허미트 곡선이나 베지어 곡선, 비유리 B-스플라인 곡선은 아핀 불변성은 성립하나 원근투영 불변성은 성립하지 않는다. 반면 NURBS 곡선은 아핀 불변성뿐 아니라 원근투영 불변성도 성립한다. 균일 B-스플라인 곡선은 제어점을 근사화 하는 근사 곡선이며, 모든 제어점을 연결하는 보간 곡선은 아니다.

교재 관련 부분 : 303~305쪽

53. 다음 중 깊이버퍼 알고리즘에 대한 올바른 설명은? (3점)

- ① 한 다각형의 앞면과 뒷면을 구분하기 위한 알고리즘이다.
- ② 프레임 버퍼와 함께 깊이 버퍼가 필요하다.
- ③ 객체 공간 가시면 결정 알고리즘이다.
- ④ 다각형을 깊이 순서에 따라 정렬한 후 주사변환을 한다.

정답 : ②

해설 : 깊이버퍼 알고리즘은 각 픽셀 단위로 현재 프레임 버퍼에 저장된 픽셀의 깊이 값에 해당되는 값을 깊이버퍼에 저장하고, 그 픽셀에 새로운 물체의 색상을 저장하려고 할 때 기존 깊이버퍼의 값보다 작을 경우만, 즉, 현재 프레임 버퍼에 그려놓은 물체보다 앞에 있는 경우만 새로운 색으로 대체하도록 한다. 이처럼 이미지의 픽셀 단위로 깊이를 비교하여 가시성을 처리하므로 이미지 공간 가시면 결정 알고리즘이라고 한다. ④는 깊이정렬기법을 의미하는데, 멀리 있는 물체부터 그리고 가까운 물체는 그 위에 덧칠하여 그리는 방식으로 화가의 알고리즘이라고도 부르는 방식이다.

교재 관련 부분 : 333, 335, 336쪽

54. 다음 중 화가의 알고리즘이라 부르는 가시면 결정 알고리즘은 무엇인가? (3점)

- ① 깊이 정렬 기법
- ② z-버퍼 기법
- ③ A-버퍼 알고리즘
- ④ 주사선 알고리즘

정답 : ①

해설 : 53번 문항 해설 참조

교재 관련 부분 : 341~344

55. 점광원에 대한 올바른 설명은? (4점)

- ① 광원으로부터의 거리와 관계없이 광량이 일정하다.
- ② 물체의 모든 면에 균일한 빛이 가해진다.
- ③ 모든 방향으로 빛을 방사하는 전방향성 광원이다.
- ④ 빛이 전달되는 방향은 평행하다.

정답 : ③

해설 : 점 광원은 이름 그대로 한 점의 위치에서 모든 방향으로 퍼져 나가는 광원이다. 따라서 빛은 방사형으로 전달된다. 점으로부터 멀리 갈수록 빛이 더 넓은 범위로 펼쳐지므로, 광량은 감소한다. ②는 주변광에 대한 설명이며, ①과 ④는 방향성 광원에 대한 설명이다.

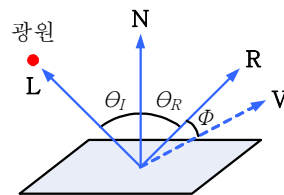
교재 관련 부분 : 363~366쪽

56. Phong 반사 모델에서 정반사에 의한 경면광의 광량을 나타내는 식은? I_s 는 입사되는 광량, k_s 는 반사율, L 은 광원방향 단위벡터, R 은 반사각방향 단위벡터, V 는 시점방향 단위벡터, n_s 는 광택계수이다. (2점)

- ① $k_s I_s (R \times V)$ ② $k_s I_s (R \cdot V)^{n_s}$
③ $k_s I_s (L \cdot R)^{n_s}$ ④ $n_s k_s I_s (L \cdot V)$

정답 : ②

해설 : Phong 반사 모델에서 정반사에 의한 경면광 광량은 다음 수식으로 정의된다.



$$I_{Spec} = k_s I_s \cos^{n_s} \phi$$

ϕ 가 0도인 방향으로 가장 강한 반사가 일어난다. ϕ 가 커질수록 반사광이 약해지는데, 광택계수 n_s 가 클수록 급격히 감소한다. 이때 $\cos \phi$ 는 단위벡터 R 과 V 의 내적이므로 이 식은 다음과 같이 표현한 것과 같다.

$$I_{Spec} = k_s I_s (R \cdot V)^{n_s}$$

교재 관련 부분 : 368~369쪽

※ (57~60) 다음 OpenGL 프로그램에 대한 질문에 답하시오. (여기에서 (㉠)에는 입체를 그리는 명령이 포함된다고 가정한다.)

```
#include <GL/glut.h>

void init()
{
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glViewport(0, 0, 300, 300);
    (㉠)
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-2.0, 2.0, -2.0, 2.0, 1.0, 5.0);
```

```

}

void wireQuadSurface()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glClear(GL_DEPTH_BUFFER_BIT);

    glEnable(GL_POLYGON_OFFSET_FILL);
    glPolygonOffset(1.0, 1.0);
    glMatrixMode(____(㉔)____);
    glLoadIdentity();
    gluLookAt(3.0, 3.0, 3.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0); // (㉕)

    _____(㉖)_____

    _____(㉗)_____

    glFlush();
}

int main(int argc, char**argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(winWidth, winHeight);
    glutCreateWindow("Depth Buffer Algorithm");

    init();
    _____(㉘)_____

    glutMainLoop();

    return 0;
}

```

57. 가시면 결정 기능을 사용하기 위해 (㉘)에 넣을 문장은? (2점)

- ① glEnable(GL_LINE_SMOOTH); ② glEnable(GL_BLEND);
 ③ glEnable(GL_LIGHT0); ④ glEnable(GL_DEPTH_TEST);

정답 : ④

해설 : 깊이버퍼 알고리즘을 적용할 수 있게 하는 문장을 사용한다.

교재 관련 부분 : 349쪽

58. 3차원 뷰잉 파라미터를 설정하는 문장 중 (ㄴ)과 (ㄷ)에 대한 설명으로 올바른 것은?
(2점)

- ① (ㄴ)에는 GL_PROJECTION을 넣는다.
- ② 원근투영을 지정하고 있다.
- ③ 뷰평면 법선벡터는 원점에서 (3, 3, 3)을 향하는 방향이다.
- ④ x 축 방향이 뷰업벡터의 방향이다.

정답 : ③

해설 : (ㄴ)은 뷰잉 파라미터를 설정하기 위한 행렬모드를 선택하기 위한 문장으로, GL_MODELVIEW 행렬모드를 선택한다. (ㄷ)은 뷰잉 파라미터를 설정하는 문장으로, (3, 3, 3) 위치에서 (0, 0, 0)을 바라보며, 뷰업벡터는 (0, 1, 0)이다. 그러므로 뷰평면 법선벡터는 (0, 0, 0)에서 (3, 3, 3)을 향하는 방향이며, 뷰업벡터는 y 축 방향이다.

교재 관련 부분 : 257쪽

59. (ㄷ)에는 (ㄷ)에 정의되는 물체를 고정점 (1, 2, 3)을 기준으로 2배 확대하는 기하변환 명령을 넣고자 한다. 다음 문장들을 (ㄷ)에 넣을 순서대로 나열한 것은? (2점)

(가) `glTranslatef(1.0f, 2.0f, 3.0f);`
(나) `glTranslatef(-1.0f, -2.0f, -3.0f);`
(다) `glScalef(2.0f, 2.0f, 2.0f);`

- ① (가) → (나) → (다) ② (가) → (다) → (나)
- ③ (나) → (다) → (가) ④ (다) → (나) → (가)

정답 : ②

해설 : 원하는 변환은 (-1, -2, -3)만큼 이동하는 변환 $T(-1, -2, -3)$ 을 한 후, 2배 확대하는 변환 $S(2, 2, 2)$ 를 하고, 이어서 다시 (1, 2, 3)만큼 이동하는 $T(1, 2, 3)$ 을 차례로 수행한다. 이를 위한 복합 변환행렬은

$$T(1, 2, 3) \cdot S(2, 2, 2) \cdot T(-1, -2, -3)$$

이다. 그런데 OpenGL에서는 이 식의 좌측 행렬부터 차례로 나열하여 곱하는 형식으로 명령을 나열해야 한다. 그러므로 $T(1, 2, 3)$ 를 위한 명령이 첫째, $S(2, 2, 2)$ 를 위한 명령이 둘째, $T(-1, -2, -3)$ 를 위한 명령이 셋째로 나열되어야 한다.

교재 관련 부분 : 178~183쪽

60. (ㄷ)에 넣을 적절한 명령은? (2점)

- ① wireQuadSurface();
- ② glutReshapeFunc();
- ③ glEnable(GL_CULL_FACE);
- ④ glutDisplayFunc(wireQuadSurface);

정답 : ④

해설 : 디스플레이를 위한 함수인 wireQuadSurface를 디스플레이 콜백함수로 등록하는
문장이 필요하다.

교재 관련 부분 : 57~59쪽