02

CHAPTER

LINEAR ALGEBRA

행렬

개요

- 행렬과 관련된 전반적인 논제들을 학습
- 행렬의 기본적인 개념과 여러 가지 연산들의 방법론을 고찰
- 대각행렬, 항등행렬, 전치행렬 등 다양한 행렬들의 종류와 특성을 살펴봄
- 행렬에 있어서의 3가지 기본 연산을 통해 행 사다리꼴로의 변형과 계수를 구하고 그 의미를 고찰함
- C프로그램과 MATLAB을 통하여 행렬의 합과 곱, 대각합, 계수 등의 결과를 비교하며 실습해 보고 학습한다.

02

CHAPTER

행렬

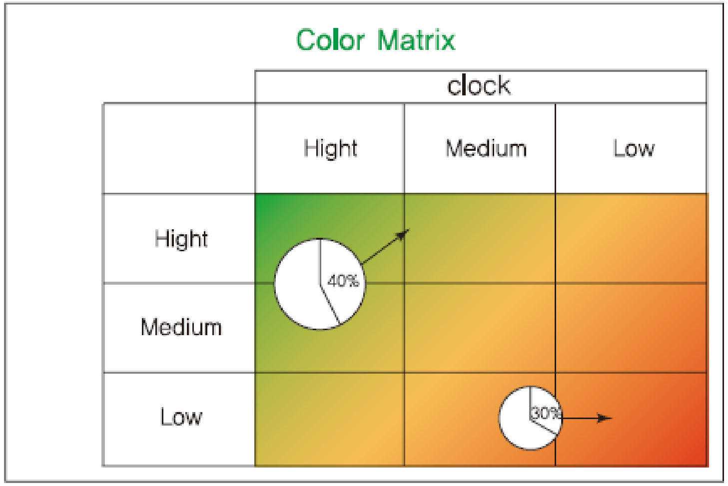
LINEAR ALGEBRA

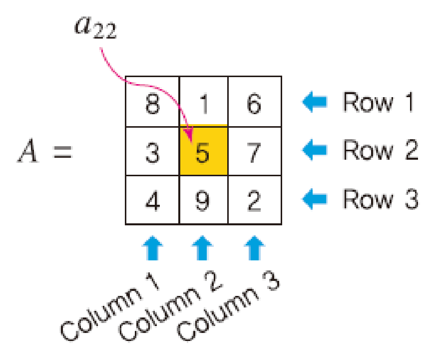
CONTENTS

2.1 행렬과 행렬의 연산

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}$$

$$M = \begin{bmatrix} 0 & 0 & \delta & 0 & 0 & 0 & 0 & \cdots \\ 0 & \varepsilon_{00} & \Delta_0 & \varepsilon_{01} & 0 & \varepsilon_{02} & 0 & \cdots \\ \delta & \Delta_0 & 0 & \Delta_1 & 0 & \Delta_2 & 0 & \cdots \\ 0 & \varepsilon_{10} & \Delta_1 & \varepsilon_{11} & m_1 & \varepsilon_{12} & 0 & \cdots \\ 0 & 0 & 0 & m_1 & 0 & 0 & 0 & \cdots \\ 0 & \varepsilon_{20} & \Delta_2 & \varepsilon_{21} & 0 & \varepsilon_{22} & m_2 & \cdots \\ 0 & 0 & 0 & 0 & 0 & m_2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$





2.1 행렬과 행렬의 연산

행렬(matrix)은 수 또는 문자를 배열의 형태로 나타내는 것을 말하는데, 그 어원은 라틴어 Mater(어머니) + ~ix의 합성어로 모체(母體)를 의미한다. m, n을 양의 정수라고 할 때 실수들로 이루어지는 다음과 같은 배열을 **행렬(matrix, 行列)**이라고 부른다.

$$A = \begin{array}{c} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{in} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \\ \uparrow \text{열} \end{array} \begin{array}{l} \leftarrow \text{행} \\ \\ \\ \\ \\ \\ \\ \end{array}$$

$$m \times n$$

이 행렬을 간단하게 $A=[a_{ij}]$, $i=1, \cdots, m$, $j=1, \cdots, n$이라 적고, $m \times n$ **행렬** 또는 (m, n) **행렬**이라고 부른다. 이 행렬은 m개의 **행**(row)과 n개의 **열**(column)을 가지고 있다. 예를 들면, 제1행은

$$[a_{11}, a_{12}, \cdots, a_{1n}]$$ 이고,

제2열은

$$\begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix}$$ 이다.

2.1 행렬과 행렬의 연산

a_{ij}를 이 행렬의 $ij-$항($ij-$entry) 또는 $ij-$성분($ij-$component)이라고 부르는데, a_{ij}는 위로부터 i번째의 행과 j번째의 열이 만나는 항의 값이다.

$$
\begin{array}{ccc}
 & \begin{array}{ccccc} \phantom{a_{11}} & & j\text{번째 열} & & \phantom{a_{1n}} \\ & & \downarrow & & \end{array} & \\
A = & \begin{bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ a_{i1} & \cdots & \boxed{a_{ij}} & \cdots & a_{in} \\ \vdots & & \vdots & & \vdots \\ a_{m1} & \cdots & a_{mj} & \cdots & a_{mn} \end{bmatrix} & \begin{array}{l} \\ \\ \leftarrow i\text{번째 행} \\ \\ \\ \end{array}
\end{array}
$$

2.1 행렬과 행렬의 연산

예제 ❷-1 다음의 행렬 A가 2×3 행렬임을 알아보자.

$$A = \begin{bmatrix} 1 & 1 & -2 \\ -1 & 4 & -5 \end{bmatrix}$$

풀이 이 행렬은 2개의 행과 3개의 열을 가진다. 행은 $[1,\ 1,\ -2]$, $[-1,\ 4,\ -5]$이고, 열은

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \begin{bmatrix} -2 \\ -5 \end{bmatrix}$$이다. ■

2.1 행렬과 행렬의 연산

이와 같이 행렬의 각 행은 가로의 n 순서쌍으로 볼 수 있고, 각 열은 세로의 m 순서쌍으로 볼 수 있다. 가로의 n 순서쌍을 **행벡터**(row vector), 세로의 m 순서쌍을 **열벡터**(column vector)라고도 부른다.

여기서 행벡터 $[x_1, \cdots, x_n]$은 $1\times n$ 행렬이고, 열벡터

$$\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

n개의 행과 n개의 열을 가지는 행렬을 **n차 정방행렬(square matrix of order n)**이라고도 하는데, 색깔로 표시된 부분의 성분 $a_{11}, a_{22}, \cdots, a_{nn}$은 A의 **주대각선(main diagonal)**상에 있다고 한다.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

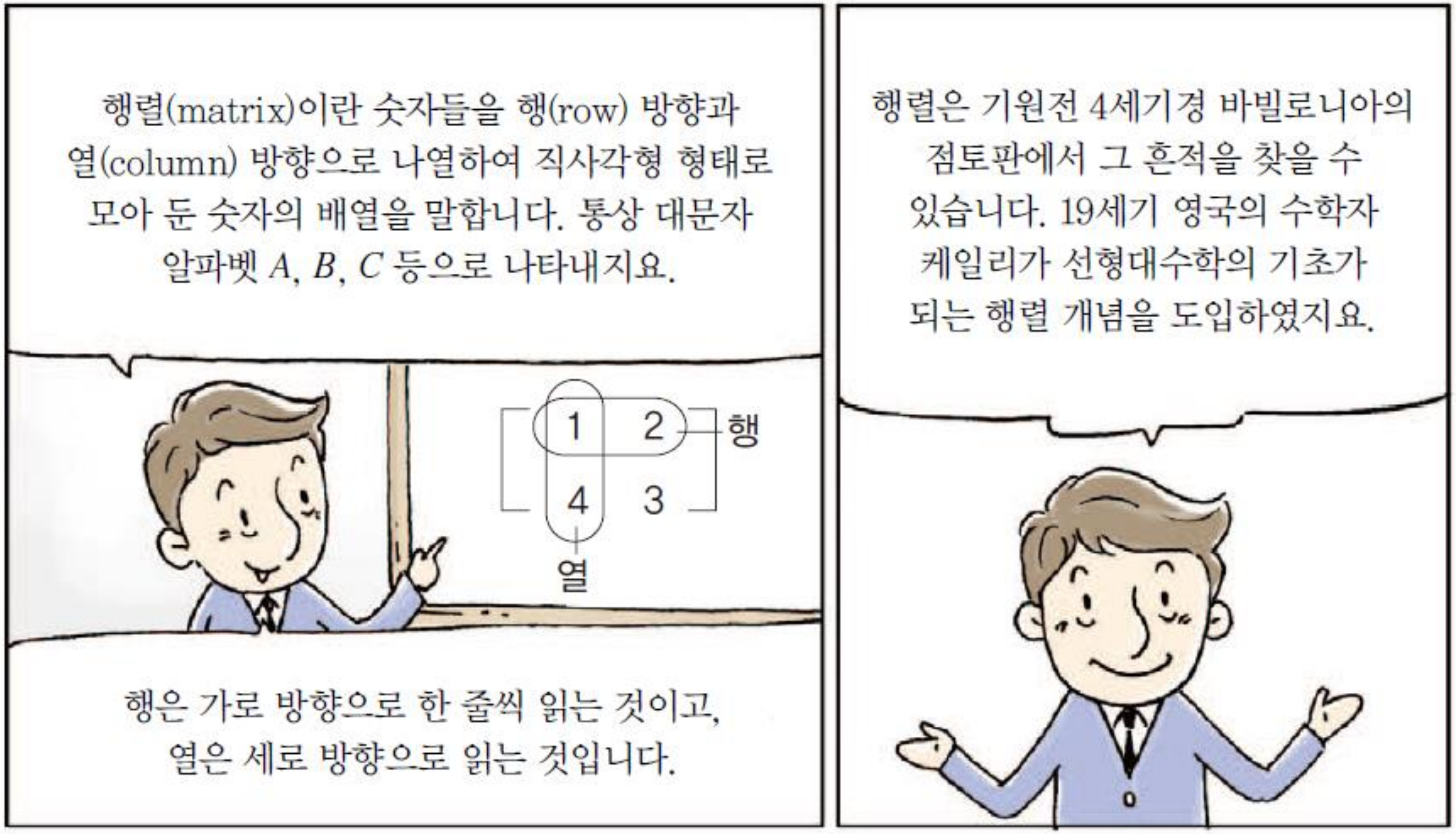
2.1 행렬과 행렬의 연산

정의 ❷-1 행렬 $A = [a_{ij}],\ i = 1, \cdots, m,\ j = 1, \cdots, n$에 대해 만일 $m = n$인 경우, 즉 행의 개수와 열의 개수가 같은 경우인 $A_{n \times n}$일 때, 이를 **정방행렬(square matrix)**이라고 한다. 예를 들면,

$$\underset{2 \times 2}{\begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix}},\quad \underset{3 \times 3}{\begin{bmatrix} 1 & -1 & 5 \\ 2 & 6 & -1 \\ 3 & 1 & -1 \end{bmatrix}},\quad \underset{4 \times 4}{\begin{bmatrix} 3 & 1 & 2 & -1 \\ 1 & 2 & 1 & 0 \\ -1 & -4 & 1 & -2 \\ 1 & -2 & 5 & -4 \end{bmatrix}}$$

은 모두 정방행렬이다.

2.1 행렬과 행렬의 연산


행렬(matrix)이란 숫자들을 행(row) 방향과 열(column) 방향으로 나열하여 직사각형 형태로 모아 둔 숫자의 배열을 말합니다. 통상 대문자 알파벳 A, B, C 등으로 나타내지요.
1 2 행
4 3
열
행은 가로 방향으로 한 줄씩 읽는 것이고, 열은 세로 방향으로 읽는 것입니다.
행렬은 기원전 4세기경 바빌로니아의 점토판에서 그 흔적을 찾을 수 있습니다. 19세기 영국의 수학자 케일리가 선형대수학의 기초가 되는 행렬 개념을 도입하였지요.

C program, MATLAB

예제 ❷-2

$A=\begin{bmatrix}3 & -1\\ 2 & 4\end{bmatrix}$, $B=\begin{bmatrix}2 & 1\\ -3 & 2\end{bmatrix}$일 때 $A+B$를 구해 보자.

풀이 $A+B=\begin{bmatrix}3+2 & (-1)+1\\ 2+(-3) & 4+2\end{bmatrix}=\begin{bmatrix}5 & 0\\ -1 & 6\end{bmatrix}$

이 된다.

또한 $A=\begin{bmatrix}3 & -1 & -5\\ 2 & 3 & 4\end{bmatrix}$, $B=\begin{bmatrix}5 & 1 & -1\\ 2 & 1 & -1\end{bmatrix}$이라 하면

$$A+B=\begin{bmatrix}3+5 & (-1)+1 & (-5)+(-1)\\ 2+2 & 3+1 & 4+(-1)\end{bmatrix}=\begin{bmatrix}8 & 0 & -6\\ 4 & 4 & 3\end{bmatrix}$$

이 된다. ■

C program, MATLAB

예제 ❷-3

다음 두 행렬 A, B의 합을 구해 보자.

$$A=\begin{bmatrix}3 & -1 & 3\\ 0 & 4 & 6\\ 2 & 7 & -5\end{bmatrix},\quad B=\begin{bmatrix}0 & -2 & -4\\ 1 & 6 & -2\\ 1 & -1 & 2\end{bmatrix}$$

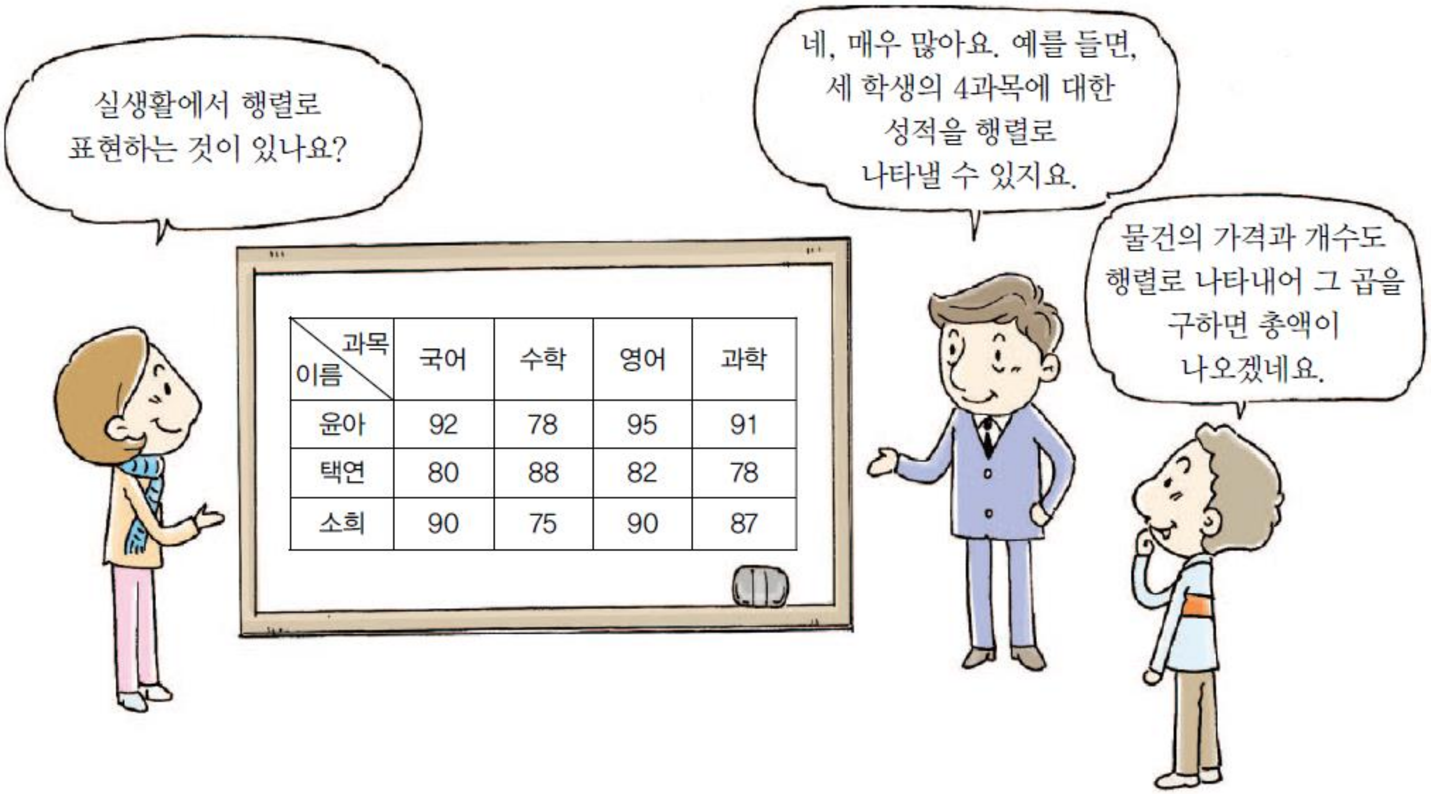
풀이 해당하는 항들을 각각 더한다.

$$A+B=\begin{bmatrix}3+0 & (-1)+(-2) & 3+(-4)\\ 0+1 & 4+6 & 6+(-2)\\ 2+1 & 7+(-1) & (-5)+2\end{bmatrix}$$

$$=\begin{bmatrix}3 & -3 & -1\\ 1 & 10 & 4\\ 3 & 6 & -3\end{bmatrix}$$

이 된다. ■

정의 ❷−2 | 행렬의 차(뺄셈)는 다음과 같이 정의될 수 있다. A와 B가 모두 $m \times n$ 행렬일 때 $A - B$ 또는 $A + (-1)B$라고 쓰고 A에서 B를 뺀 차(difference)라고 한다.



이름 \ 과목	국어	수학	영어	과학
윤아	92	78	95	91
택연	80	88	82	78
소희	90	75	90	87

행렬과 행렬의 연산

정의 ❷-3

행렬의 스칼라 곱은 다음과 같이 정의된다. k가 실수 값이고, $A = [a_{ij}]$를 임의의 행렬이라고 할 때, kA는 ij–성분이 ka_{ij}인 행렬로 정의되므로 $kA = [ka_{ij}]$이다.

즉, 행렬 A에다 스칼라 값 k를 곱했을 때 $k \cdot A$는 행렬 A의 각 항에다 k를 곱함으로써 얻어지므로 다음과 같은 행렬로 표현할 수 있다.

$$kA = \begin{bmatrix} ka_{11} & ka_{12} & \cdots & ka_{1n} \\ ka_{21} & ka_{22} & \cdots & ka_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ ka_{m1} & ka_{m2} & \cdots & ka_{mn} \end{bmatrix}$$

예제 ❷-4 행렬 A, B가 다음과 같을 때 스칼라 곱을 구해 보자.

$$A=\begin{bmatrix} 1 & -1 & 4 \\ -2 & 3 & 2 \end{bmatrix},\quad B=\begin{bmatrix} 3 & 1 & -1 \\ -2 & 0 & 2 \end{bmatrix}$$

풀이 스칼라 값 c, d를 각각 $c=2$, $d=-3$이라고 하면

$$cA=2A=\begin{bmatrix} 2 & -2 & 8 \\ -4 & 6 & 4 \end{bmatrix}$$

$$dB=-3B=\begin{bmatrix} -9 & -3 & 3 \\ 6 & 0 & -6 \end{bmatrix}$$

이다. 또한

$$(-1)A=-A=\begin{bmatrix} -1 & 1 & -4 \\ 2 & -3 & -2 \end{bmatrix}$$

이다. ■

예제 ❷-5 $A=\begin{bmatrix}1 & -2 & 3\\ 0 & 4 & 5\end{bmatrix}$이고 $B=\begin{bmatrix}4 & 6 & 8\\ 1 & -3 & -7\end{bmatrix}$일 때 $2A-3B$를 구해 보자.

풀이 $$2A-3B=\begin{bmatrix}2 & -4 & 6\\ 0 & 8 & 10\end{bmatrix}+\begin{bmatrix}-12 & -18 & -24\\ -3 & 9 & 21\end{bmatrix}$$

$$=\begin{bmatrix}-10 & -22 & -18\\ -3 & 17 & 31\end{bmatrix}$$

이다. ■

정의 ❷-4 | 두 개의 $m\times n$ 행렬 $A=[a_{ij}]$와 $B=[b_{ij}]$에서 만약 대응하는 항들이 동일하다면, 즉 $i=1, 2, \cdots, m$, $j=1, 2, \cdots, n$인 경우 $a_{ij}=b_{ij}$라면 **상등(equal)**하다고 한다.

예제 ❷-6 다음의 행렬 관계가 주어졌을 때 x, y, z, t의 값을 구해 보자.

$$3\begin{bmatrix} x & y \\ z & t \end{bmatrix} = \begin{bmatrix} x & 6 \\ -1 & 2t \end{bmatrix} + \begin{bmatrix} 4 & x+y \\ z+t & 3 \end{bmatrix}$$

풀이 왼쪽과 오른쪽의 식을 정리하면

$$\begin{bmatrix} 3x & 3y \\ 3z & 3t \end{bmatrix} = \begin{bmatrix} x+4 & x+y+6 \\ z+t-1 & 2t+3 \end{bmatrix}$$

대응하는 항들을 같게 놓고 나면 다음과 같은 식을 얻을 수 있다.

$$3x = x + 4,\ 3y = x + y + 6,\ 3z = z + t - 1,\ 3t = 2t + 3$$

이것을 풀면

$$2x = 4,\ 2y = x + 6,\ 2z = t - 1,\ t = 3$$

따라서

$$x = 2, y = 4,\ z = 1,\ t = 3$$

이라는 값을 얻는다. ■

정리 ❷-1 행렬의 합과 스칼라 곱은 같은 크기의 행렬 A, B, C와 어떤 상수 c, d가 주어졌을 때 다음과 같은 연산법칙들을 만족한다. 여기서 O은 모든 항들이 0인 영행렬이다.

(1) $A+B=B+A$ (덧셈의 교환법칙)

(2) $(A+B)+C=A+(B+C)$ (덧셈의 결합법칙)

(3) $A+O=O+A$ (덧셈의 항등법칙)

(4) $A+(-A)=(-A)+A=O$ (덧셈의 역원)

(5) $c(A+B)=cA+cB$ (스칼라 곱의 배분법칙)

(6) $(c+d)A=cA+dA$ (스칼라 곱의 배분법칙)

증명 (1) 행렬의 합의 정의에 의하여 $A+B$와 $B+A$는 둘 다 $m \times n$ 행렬이다. 모든 (i, j)에 대해

$$[A+B]_{ij}=[A]_{ij}+[B]_{ij}=[B]_{ij}+[A]_{ij}=[B+A]_{ij}$$

따라서 $A+B=B+A$이다.

(2) 같은 맥락으로 $A+(B+C)$와 $(A+B)+C$는 둘 다 $m\times n$ 행렬이다. 모든 (i, j)에 대해

$$\begin{aligned}[A+(B+C)]_{ij} &= A_{ij}+[B+C]_{ij} \\ &= A_{ij}+(B_{ij}+C_{ij}) \\ &= (A_{ij}+B_{ij})+C_{ij} \\ &= [A+B]_{ij}+C_{ij} \\ &= [(A+B)+C]_{ij}\end{aligned}$$

따라서 행렬의 덧셈에 대하여 결합법칙이 성립한다. ■

2.1 행렬과 행렬의 연산

정의 ❷−5 $A=[a_{ij}]$가 $m\times n$ 행렬이고, $B=[b_{ij}]$가 $n\times p$ 크기의 행렬일 때 행렬 A와 B의 행렬의 곱(multiplication)은 $AB=C=[c_{ij}]$로써 다음과 같이 정의되는 $m\times p$ 행렬이 된다.

$$c_{ij}=\sum_{k=1}^{n} a_{ik}b_{kj}=a_{i1}b_{1j}+a_{i2}b_{2j}+\cdots+a_{in}b_{nj}$$

$$i=1, 2, \cdots, m$$

$$j=1, 2, \cdots, p$$

여기서 주목할 점은 AB는 A의 열의 숫자가 B의 행의 숫자와 같을 경우에만 정의된다는 점이다. 또한 C의 (i, j)항들은 A의 i번째 행과 B의 j번째 열을 곱한 합으로부터 만들어진다는 점에 유의한다.

$$AB = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{in} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1j} & \cdots & b_{1q} \\ b_{21} & b_{22} & \cdots & b_{2j} & \cdots & b_{2q} \\ \vdots & \vdots & & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nj} & \cdots & b_{nq} \end{bmatrix}$$

$$m \times n \qquad\qquad n \times p$$

$$= \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & c_{ij} & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{bmatrix} = C$$

$$m \times p$$

예제 ❷-7 다음 행렬 A, B의 곱셈이 정의되는지를 판별하고 곱셈이 가능하다면 그 값을 구해 보자.

$$A = \begin{bmatrix} -4 & 2 \\ 1 & 6 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 3 \\ -2 \\ 7 \end{bmatrix}$$

풀이 앞의 행렬은 3×2 행렬이고, 뒤의 행렬은 3×1 행렬이다. 따라서 내측의 2와 3이 다르기 때문에 행렬의 곱이 성립될 수 없다. ■

예제 ❷-8

두 행렬이 각각 다음과 같은 크기의 행렬이라고 할 때, 두 행렬의 곱이 나올 수 있는 행렬의 크기를 각각 구해 보자.

(1) $(2\times3)(3\times4)$ (2) $(1\times2)(3\times1)$

(3) $(4\times4)(3\times3)$ (4) $(4\times1)(1\times2)$

(5) $(5\times2)(2\times3)$ (6) $(2\times2)(2\times4)$

풀이 중간의 숫자가 일치할 때 행렬의 곱이 정의될 수 있으며, 그 결과는 제일 앞의 숫자와 제일 뒤의 숫자로 나타난다. 따라서 그 결과는 다음과 같다.

(1) 2×4 (2) 정의될 수 없다.

(3) 정의될 수 없다. (4) 4×2

(5) 5×3 (6) 2×4 ■

2.1 행렬과 행렬의 연산

C program, MATLAB

예제 ❷-9

행렬 A와 행렬 B가 다음과 같이 주어졌을 때, 행렬의 곱 AB의 두 번째 행의 값을 구해 보자.

$$A=\begin{bmatrix}2&4&-1\\-1&3&3\\4&-2&1\\-3&0&2\end{bmatrix},\quad B=\begin{bmatrix}4&-2\\-2&1\\3&-1\end{bmatrix}$$

풀이 A는 4×3 행렬이고 B는 3×2 행렬이므로 곱 AB는 4×2 행렬이 된다. 행렬 간 곱셈의 행과 열의 규칙에 따라 행렬 곱의 2행의 항들은 A의 2행과 B의 열들의 곱으로부터 다음과 같이 계산될 수 있다.

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

$$=\begin{bmatrix}\square&\square\\-4-6+9&2+3-3\\\square&\square\\\square&\square\end{bmatrix}=\begin{bmatrix}\square&\square\\-1&2\\\square&\square\\\square&\square\end{bmatrix}$$

이와 같은 방법으로 나머지 행들의 값들을 구하면

$$\begin{bmatrix} -3 & 1 \\ -1 & 2 \\ 23 & -11 \\ -6 & 4 \end{bmatrix}$$

이 된다. ■

C program, MATLAB

예제 ❷-10

행렬 A, B, C가 다음과 같이 주어졌을 때, AB와 AC를 구해 보자.

$$A=\begin{bmatrix}1 & 1 & 0\\ 2 & 0 & 1\end{bmatrix},\quad B=\begin{bmatrix}2 & 0\\ 0 & 1\\ 1 & 3\end{bmatrix},\quad C=\begin{bmatrix}1 & -2\\ -1 & 2\\ -2 & 4\end{bmatrix}$$

풀이 (1) $AB=\begin{bmatrix}1 & 1 & 0\\ 2 & 0 & 1\end{bmatrix}\begin{bmatrix}2 & 0\\ 0 & 1\\ 1 & 3\end{bmatrix}=\begin{bmatrix}\text{row1}\times\text{col1} & \text{row1}\times\text{col2}\\ \text{row2}\times\text{col1} & \text{row2}\times\text{col2}\end{bmatrix}$

$$=\begin{bmatrix}(1)(2)+(1)(0)+(0)(1) & (1)(0)+(1)(1)+(0)(3)\\ (2)(2)+(0)(0)+(1)(1) & (2)(0)+(0)(1)+(1)(3)\end{bmatrix}=\begin{bmatrix}2 & 1\\ 5 & 3\end{bmatrix}$$

(2) $AC=\begin{bmatrix}1 & 1 & 0\\ 2 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & -2\\ -1 & 2\\ -2 & 4\end{bmatrix}=\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$ ■

예제 ❷-11 다음 행렬의 곱을 구해 보자.

(1) $\begin{bmatrix} 1 & 2 & -1 \\ 0 & -5 & 3 \end{bmatrix}\begin{bmatrix} 4 \\ 3 \\ 7 \end{bmatrix}$ (2) $\begin{bmatrix} 2 & -3 \\ 8 & 0 \\ -5 & 2 \end{bmatrix}\begin{bmatrix} 4 \\ 7 \end{bmatrix}$

(3) $\begin{bmatrix} 1 & 2 & -1 \\ 3 & 1 & 4 \end{bmatrix}\begin{bmatrix} -2 & 5 \\ 4 & -3 \\ 2 & 1 \end{bmatrix}$ (4) $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r \\ s \\ t \end{bmatrix}$

풀이 행과 열의 곱에 따라 두 행렬의 곱의 결과는 다음과 같다.

$$(1)\ \begin{bmatrix} 1 & 2 & -1 \\ 0 & -5 & 3 \end{bmatrix}\begin{bmatrix} 4 \\ 3 \\ 7 \end{bmatrix} = \begin{bmatrix} 1\cdot 4+2\cdot 3+(-1)\cdot 7 \\ 0\cdot 4+(-5)\cdot 3+3\cdot 7 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$$

$$(2)\ \begin{bmatrix} 2 & -3 \\ 8 & 0 \\ -5 & 2 \end{bmatrix}\begin{bmatrix} 4 \\ 7 \end{bmatrix} = \begin{bmatrix} 2\cdot 4+(-3)\cdot 7 \\ 8\cdot 4+0\cdot 7 \\ (-5)\cdot 4+2\cdot 7 \end{bmatrix} = \begin{bmatrix} -13 \\ 32 \\ -6 \end{bmatrix}$$

$$(3)\ \begin{bmatrix}1 & 2 & -1\\ 3 & 1 & 4\end{bmatrix}\begin{bmatrix}-2 & 5\\ 4 & -3\\ 2 & 1\end{bmatrix}=$$

$$\begin{bmatrix}1\cdot(-2)+2\cdot 4+(-1)\cdot 2 & 1\cdot 5+2\cdot(-3)+(-1)\cdot 1\\ 3\cdot(-2)+1\cdot 4+4\cdot 2 & 3\cdot 5+1\cdot(-3)+4\cdot 1\end{bmatrix}$$

$$=\begin{bmatrix}4 & -2\\ 6 & 16\end{bmatrix}$$

$$(4)\ \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}r\\ s\\ t\end{bmatrix}=\begin{bmatrix}1\cdot r+0\cdot s+0\cdot t\\ 0\cdot r+1\cdot s+0\cdot t\\ 0\cdot r+0\cdot s+1\cdot t\end{bmatrix}=\begin{bmatrix}r\\ s\\ t\end{bmatrix}\quad\blacksquare$$

C program, MATLAB

예제 ❷-12

다음의 두 행렬 A, B가 주어졌을 때, 이 행렬들의 곱셈에서는 교환법칙이 성립하지 않음을 살펴보자. 즉, $AB \neq BA$임을 보인다.

$$A = \begin{bmatrix} 5 & 1 \\ 3 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 4 & 3 \end{bmatrix}$$

풀이 AB와 BA의 값이 서로 다르다는 것을 보인다.

$$AB = \begin{bmatrix} 5 & 1 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 14 & 3 \\ -2 & -6 \end{bmatrix}$$

$$BA = \begin{bmatrix} 2 & 0 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 5 & 1 \\ 3 & -2 \end{bmatrix} = \begin{bmatrix} 10 & 2 \\ 29 & -2 \end{bmatrix}$$

따라서 $AB \neq BA$이다. ■

2.1 행렬과 행렬의 연산

여기서 잠깐 !!

행렬에서 덧셈의 교환법칙은 성립한다. 즉, $A+B=B+A$이다. 그러나 곱셈에서는 교환법칙이 일반적으로 성립되지 않는다. 예를 들어,

$$AB=\begin{bmatrix}1&0\\1&0\end{bmatrix}\begin{bmatrix}0&0\\1&1\end{bmatrix}=\begin{bmatrix}0&0\\0&0\end{bmatrix}$$

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

이다.

따라서 $\begin{bmatrix}0&0\\0&0\end{bmatrix}\neq\begin{bmatrix}0&0\\2&0\end{bmatrix}$이므로 $AB\neq BA$이다.

예제 ❷-13 $\begin{bmatrix} 6 & 5 \\ -4 & -3 \\ 7 & 6 \end{bmatrix}\begin{bmatrix} 2 \\ -3 \end{bmatrix}$의 값을 구하는 방법은 2가지가 있음을 예를 들어 살펴보자.

(1) 앞의 곱셈 방법으로 풀면 다음과 같다.

$$\begin{bmatrix} 6 & 5 \\ -4 & -3 \\ 7 & 6 \end{bmatrix}\begin{bmatrix} 2 \\ -3 \end{bmatrix} = \begin{bmatrix} 6\cdot 2 + 5\cdot(-3) \\ (-4)\cdot 2 + (-3)\cdot(-3) \\ 7\cdot 2 + 6\cdot(-3) \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ -4 \end{bmatrix}$$

(2) 다른 방법은 다음과 같이 풀 수 있으며, 그 결과는 항상 같다.

$$\begin{bmatrix} 6 & 5 \\ -4 & -3 \\ 7 & 6 \end{bmatrix}\begin{bmatrix} 2 \\ -3 \end{bmatrix} = 2\begin{bmatrix} 6 \\ -4 \\ 7 \end{bmatrix} - 3\begin{bmatrix} 5 \\ -3 \\ 6 \end{bmatrix} = \begin{bmatrix} 12 \\ -8 \\ 14 \end{bmatrix} + \begin{bmatrix} -15 \\ 9 \\ -18 \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ -4 \end{bmatrix}$$ ■

2.1 행렬과 행렬의 연산

정리 ❷-2 A가 $m \times n$ 행렬이고, B와 C는 행렬의 합과 곱에서 정의된 크기를 만족한다고 가정하고 k가 어떤 스칼라 값일 때 다음의 식들이 성립한다.

(1) $A(BC) = (AB)C$	(곱셈의 결합법칙)
(2) $A(B+C) = AB + AC$	(왼쪽 배분법칙)
(3) $(B+C)A = BA + CA$	(오른쪽 배분법칙)
(4) $k(AB) = (kA)B = A(kB)$	(스칼라 곱)
(5) $I_n A = A = AI_n$	(행렬 곱셈의 항등식)

여기서 잠깐!! 집합에서나 행렬의 덧셈에서도 결합법칙과 배분법칙이 성립하고, 행렬의 곱셈에서도 결합법칙과 배분법칙이 성립한다. 그러나 행렬의 곱셈에서의 교환법칙은 성립하지 않는다.

예제 ❷-14 다음 행렬에 대하여 곱셈에 관한 결합법칙이 성립하는지를 알아보자.

$$A=\begin{bmatrix}-1 & 3\\ 4 & -2\\ 5 & 0\end{bmatrix},\quad B=\begin{bmatrix}-3 & 2\\ -4 & 1\end{bmatrix},\quad C=\begin{bmatrix}1 & 0\\ 2 & 3\end{bmatrix}$$

풀이 행렬 간의 곱셈에서의 결합법칙이 성립하는 것을 보이기 위해 $(AB)C=A(BC)$임을 보인다.

$$AB=\begin{bmatrix}-1 & 3\\ 4 & -2\\ 5 & 0\end{bmatrix}\begin{bmatrix}-3 & 2\\ -4 & 1\end{bmatrix}=\begin{bmatrix}-9 & 1\\ -4 & 6\\ -15 & 10\end{bmatrix}\text{이므로}$$

$$(AB)C=\begin{bmatrix}-9 & 1\\ -4 & 6\\ -15 & 10\end{bmatrix}\begin{bmatrix}1 & 0\\ 2 & 3\end{bmatrix}=\begin{bmatrix}-7 & 3\\ 8 & 18\\ 5 & 30\end{bmatrix}\text{이다.}$$

또한

$$BC = \begin{bmatrix} -3 & 2 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 6 \\ -2 & 3 \end{bmatrix} \text{이므로}$$

$$A(BC) = \begin{bmatrix} -1 & 3 \\ 4 & -2 \\ 5 & 0 \end{bmatrix} \begin{bmatrix} 1 & 6 \\ -2 & 3 \end{bmatrix} = \begin{bmatrix} -7 & 3 \\ 8 & 18 \\ 5 & 30 \end{bmatrix} \text{이 된다.}$$

따라서 $(AB)C = A(BC)$이다. 그러므로 곱셈에서의 결합법칙이 성립한다. ■

정의 ❷-6 | A가 $n\times n$ 행렬이고 k가 양의 정수라고 할 때 A^k은 A를 k번 곱하는 것이다. 즉, $A^k=\underbrace{A\cdots A}_{k}$를 **행렬의 거듭제곱(Powers of a Matrix)**이라고 한다.

$$A^0=I \qquad A^k=A^{k-1}\cdot A$$

여기서 I는 항등행렬인데, 주대각선의 항들은 모두 1이고 나머지는 모두 0인 정방행렬을 의미한다(p.81, 정의 ❷-10 참조).

예제 ❷−15 다음의 행렬 A가 주어졌을 때, 그것의 거듭제곱을 구해 보자.

$$A=\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$$

풀이 $A^2=\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}=\begin{bmatrix}2 & 2\\2 & 2\end{bmatrix}$이고

$A^3=AAA=AA^2=\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}2 & 2\\2 & 2\end{bmatrix}=\begin{bmatrix}4 & 4\\4 & 4\end{bmatrix}$이다.

일반적인 해는 다음과 같다.

$$A^n=\begin{bmatrix}2^{n-1} & 2^{n-1}\\2^{n-1} & 2^{n-1}\end{bmatrix}$$ ■

예제 ❷−16 $A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$이고 $B=\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$일 때, $4A+2X=B$를 만족하는 행렬 X를 구해 보자.

풀이 방정식에서 $2X$에 관하여 정리하여 그 값을 구하고, 관계식으로부터 X의 값을 구할 수 있다.

$$\begin{aligned}2X=-4A+B&=-4\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}+\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}\\ &=\begin{bmatrix}-4 & -8\\ -12 & -16\end{bmatrix}+\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}\\ &=\begin{bmatrix}1 & -2\\ -5 & -8\end{bmatrix}\end{aligned}$$

따라서

$$X=\frac{1}{2}\begin{bmatrix}1 & -2\\ -5 & -8\end{bmatrix}=\begin{bmatrix}\frac{1}{2} & -1\\ -\frac{5}{2} & -4\end{bmatrix}$$

이 된다. ■

예제 ❷-17 다음에서 행렬 A와 x가 주어졌을 때 Ax를 구해 보자.

$$A=\begin{bmatrix} 2 & 3 & 4 \\ -1 & 5 & -3 \\ 6 & -2 & 8 \end{bmatrix},\quad x=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

풀이 2가지 방법으로 구할 수 있는데 그 결과는 언제나 같다.

(1) 앞에서 예를 든 방법으로 한 행과 한 열의 곱으로 구하는 방법이다.

$$\begin{bmatrix} 2 & 3 & 4 \\ -1 & 5 & -3 \\ 6 & -2 & 8 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 2x_1+3x_2+4x_3 \\ -x_1+5x_2-3x_3 \\ 6x_1-2x_2+8x_3 \end{bmatrix}$$

(2) 열을 중심으로 구하는 방법이다.

$$\begin{bmatrix} 2 & 3 & 4 \\ -1 & 5 & -3 \\ 6 & -2 & 8 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_1\begin{bmatrix} 2 \\ -1 \\ 6 \end{bmatrix} + x_2\begin{bmatrix} 3 \\ 5 \\ -2 \end{bmatrix} + x_3\begin{bmatrix} 4 \\ -3 \\ 8 \end{bmatrix}$$

$$= \begin{bmatrix} 2x_1 \\ -x_1 \\ 6x_1 \end{bmatrix} + \begin{bmatrix} 3x_2 \\ 5x_2 \\ -2x_2 \end{bmatrix} + \begin{bmatrix} 4x_3 \\ -3x_3 \\ 8x_3 \end{bmatrix}$$

$$= \begin{bmatrix} 2x_1 + 3x_2 + 4x_3 \\ -x_1 + 5x_2 - 3x_3 \\ 6x_1 - 2x_2 + 8x_3 \end{bmatrix} \blacksquare$$

2.1 행렬과 행렬의 연산

정의 ❷−7 행렬 A는 다음과 같은 부행렬(submatrix)들로 분할될 수 있다.

$$A = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array}\right] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

여기서

$$A_{11} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad A_{12} = \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{bmatrix}$$

$$A_{21} = \begin{bmatrix} a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix}, \quad A_{22} = \begin{bmatrix} a_{33} & a_{34} \\ a_{43} & a_{44} \end{bmatrix}$$

예제 ❷-18 행렬의 분할을 통해 다음 행렬곱을 구해 보자.

$$A=\begin{bmatrix}1&0&1&0\\0&2&3&-1\\2&0&-4&0\\0&1&0&3\end{bmatrix},\quad B=\begin{bmatrix}2&0&0&1&1&-1\\0&1&1&-1&2&2\\1&3&0&0&1&0\\-3&-1&2&1&0&-1\end{bmatrix}$$

풀이

$$A=\left[\begin{array}{cc:cc}1&0&1&0\\0&2&3&-1\\\hdashline 2&0&-4&0\\0&1&0&3\end{array}\right]=\begin{bmatrix}A_{11}&A_{12}\\A_{21}&A_{22}\end{bmatrix}$$

$$B=\left[\begin{array}{ccc:ccc}2&0&0&1&1&-1\\0&1&1&-1&2&2\\\hdashline 1&3&0&0&1&0\\-3&-1&2&1&0&-1\end{array}\right]=\begin{bmatrix}B_{11}&B_{12}\\B_{21}&B_{22}\end{bmatrix}$$

$$C=AB=\begin{bmatrix}C_{11}&C_{12}\\C_{21}&C_{22}\end{bmatrix}$$

여기서 C_{11}은 $A_{11}B_{11}+A_{12}B_{21}$이 된다. 즉

$$\begin{aligned} C_{11} &= A_{11}B_{11}+A_{12}B_{21} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 3 & -1 \end{bmatrix}\begin{bmatrix} 1 & 3 & 0 \\ -3 & -1 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 3 & 0 \\ 6 & 10 & -2 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 3 & 0 \\ 6 & 12 & 0 \end{bmatrix} \end{aligned}$$

이와 같이 C_{12}, C_{21}, C_{22}를 계산하여 적용하면 다음과 같다.

$$AB = C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \left[\begin{array}{rrr:rrr} 3 & 3 & 0 & 1 & 2 & -1 \\ 6 & 12 & 0 & -3 & 7 & 5 \\ \hdashline 0 & -12 & 0 & 2 & -2 & -2 \\ -9 & -2 & 7 & 2 & 2 & -1 \end{array}\right] \blacksquare$$

2.2 특수한 행렬

정의 ❷-8 | $n \times n$ 정방행렬에서 대각선을 제외한 모든 항들이 0인 행렬 D를 **대각행렬(diagonal matrix)**이라고 한다.

$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$$

예를 들면,

$$\begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}\text{과 } \begin{bmatrix} 4 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 2 \end{bmatrix}\text{은 둘 다 대각행렬이다.}$$

2.2 특수한 행렬

정의 ❷-9 | 정방행렬 A의 주대각선 위의 모든 성분들을 **대각항**이라고 하고, 각 대각항의 합을 **대각합(trace)**이라고 하며 $\mathrm{tr}(A)$ 또는 $\mathrm{trace}(A)$로 표기한다. 즉, 행렬의 대각합은 행과 열 번호가 같은 성분들의 합이다.

예를 들면,

$$A=\begin{bmatrix}-5 & 3\\ 4 & 2\end{bmatrix}\text{와 } B=\begin{bmatrix}b_{11} & b_{12} & b_{13}\\ b_{21} & b_{22} & b_{23}\\ b_{31} & b_{32} & b_{33}\end{bmatrix}\text{에 대해}$$

$$\mathrm{tr}(A)=(-5)+2=-3\text{이고 } \mathrm{tr}(B)=b_{11}+b_{22}+b_{33}\text{이다.}$$

MATLAB

예제 ❷-19

다음 행렬 A, B, C에서 대각항과 대각합을 구해 보자.

$$(1)\ A=\begin{bmatrix}1 & 3 & 6\\ 2 & -5 & 8\\ 4 & -2 & 9\end{bmatrix} \qquad (2)\ B=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix} \qquad (3)\ C=\begin{bmatrix}1 & 2 & -3\\ 4 & -5 & 6\end{bmatrix}$$

풀이 (1) 대각항 $=1, -5, 9$ $\quad \mathrm{tr}(A)=1-5+9=5$

(2) 대각항 $=1, 4$ $\quad \mathrm{tr}(B)=1+4=5$

(3) 대각항과 대각합은 정방행렬에서만 정의되는데 C는 정방행렬이 아니다. ■

정리 ❷-3 A와 B가 같은 크기의 정방행렬일 때 대각합은 다음의 특성들을 가진다.

(1) $\mathrm{tr}(A^T) = \mathrm{tr}(A)$

(2) $\mathrm{tr}(cA) = c\,\mathrm{tr}(A)$

(3) $\mathrm{tr}(A+B) = \mathrm{tr}(A) + \mathrm{tr}(B)$

(4) $\mathrm{tr}(A-B) = \mathrm{tr}(A) - \mathrm{tr}(B)$

(5) $\mathrm{tr}(AB) = \mathrm{tr}(BA)$

2.2 특수한 행렬

정의 ❷-10

대각행렬이면서 대각선의 항들이 모두 1인 $n \times n$ 행렬을 **항등행렬(identity matrix)** 또는 **단위행렬**이라고 한다. 행렬의 크기가 $n \times n$인 항등행렬을 통상 I_n으로 나타내는데, 문맥상 행렬의 크기가 분명할 경우에는 그냥 I로 나타내기도 한다. $n \times n$ 항등행렬의 중요한 성질은 $AI_n = A = I_n A$이다.

다음의 행렬들은 각각 I_2, I_3, I_4인 항등행렬이다.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

정의 ❷−11 성분이 모두 0인 행렬, 즉 모든 i, j에 대하여 $a_{ij} = 0$인 행렬을 **영행렬(zero matrix)**이라고 한다. 영행렬은 간단히 굵은 체의 O이라고 나타낸다. 만일 그 크기를 강조할 필요가 있는 경우에는 $m \times n$ 영행렬을 $O_{m\times n}$으로 표기하기도 한다. 다음의 행렬들은 모두 영행렬이다.

$$[0],\quad \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\quad \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

2.2 특수한 행렬

정리 ❷-4 O를 영행렬이라고 하고, 행렬 A가 이 영행렬과 크기가 같은 임의의 행렬일 경우 $A+O=O+A=A$인 관계가 성립한다.

증명 O의 항이 모두 0이므로 $A+O=O+A=A$인 관계가 성립하는 것은 명백하다.

2.2 특수한 행렬

정의 ❷-12 임의의 행렬 A에 대하여 $-A$는 행렬 $(-a_{ij})$로 정의한다. 실수에서 $a_{ij}-a_{ij}=0$이 되듯이 행렬에서도 마찬가지로 $A+(-A)=O$인 관계가 성립한다. 행렬 $-A$는 행렬 A의 **덧셈에 대한 역원** 또는 **덧셈의 역(additive inverse)**이라고 부른다.

여러 가지 행렬들의 예 1

정방행렬 $\underset{3\times 3}{\begin{bmatrix} 1 & -1 & 5 \\ 2 & 1 & -1 \\ 3 & 1 & -1 \end{bmatrix}}$ 정사각형 형태로 행과 열의 크기가 같은 행렬

대각행렬 $\begin{bmatrix} 4 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 2 \end{bmatrix}$ 주대각선만 빼고는 모든 성분이 0으로만 된 행렬

항등행렬 $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 대각행렬 중 주대각선이 모두 1이고 나머지는 0인 행렬

영행렬 $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ 모든 성분이 0으로만 된 행렬

정의 ❷-13

행렬 $A=[a_{ij}]$를 $m\times n$ 행렬이라고 할 때, $b_{ij}=a_{ji}$가 되는 $n\times m$ 행렬 $B=[b_{ij}]$를 A의 **전치행렬(transpose matrix)**이라 하고 A^T로 나타낸다. 다시 말하면 어떤 행렬의 전치행렬은 주어진 행렬의 행과 열을 서로 바꾼 행렬이 된다. 즉,

$$A=\underset{m\times n}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}} \text{일 때, } A^T=\underset{n\times m}{\begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}} \text{이다.}$$

예제 ❷-20 다음 행렬들의 전치행렬을 살펴보자.

풀이 $A=\begin{bmatrix}2 & 1 & 0\\ 1 & 3 & 5\end{bmatrix}$이면 $A^T=\begin{bmatrix}2 & 1\\ 1 & 3\\ 0 & 5\end{bmatrix}$가 되고,

$A=[2,\ \ 1,\ \ -4]$가 1×3 행벡터이면

$A^T=\begin{bmatrix}2\\ 1\\ -4\end{bmatrix}$는 3×1 열벡터가 된다. ■

예제 ❷-21 다음 행렬들의 전치행렬을 각각 구해 보자.

$$A=\begin{bmatrix}a & b\\ c & d\end{bmatrix},\qquad B=\begin{bmatrix}1 & 2 & -1\\ -3 & 2 & 7\end{bmatrix}$$

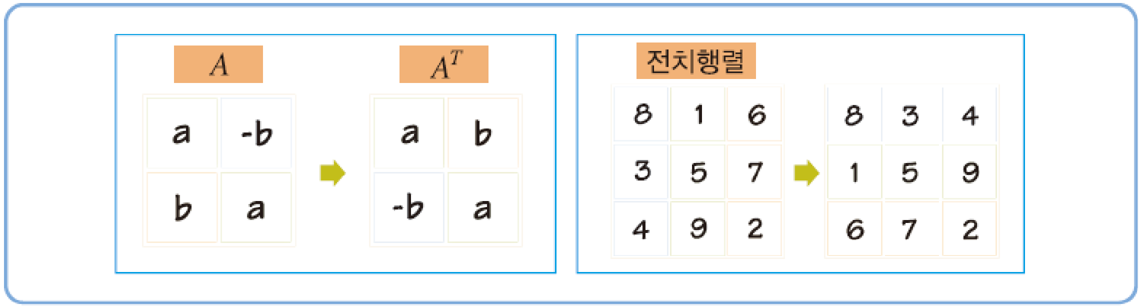
$$C=\begin{bmatrix}-5 & 2\\ 1 & -3\\ 0 & 4\end{bmatrix},\quad D=\begin{bmatrix}1 & 1 & 1 & 1\\ -3 & 5 & -2 & 7\end{bmatrix}$$

풀이 각 행렬에서 행과 열을 교환함으로써 전치행렬을 구할 수 있다.

특수한 행렬

$$A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}, \qquad B^T = \begin{bmatrix} 1 & -3 \\ 2 & 2 \\ -1 & 7 \end{bmatrix}$$

$$C^T = \begin{bmatrix} -5 & 1 & 0 \\ 2 & -3 & 4 \end{bmatrix}, \quad D^T = \begin{bmatrix} 1 & -3 \\ 1 & 5 \\ 1 & -2 \\ 1 & 7 \end{bmatrix} \blacksquare$$



2.2 특수한 행렬

정리 ❷-5 전치행렬에서는 다음과 같은 성질들이 성립한다.

(1) $(A^T)^T = A$

(2) $(A \pm B)^T = A^T \pm B^T$

(3) $(cA)^T = cA^T$(c는 상수)

(4) $(AB)^T = B^T A^T$

증명 여기서는 대표로 (1)과 (3)의 경우에 대해서만 증명한다.

(1) A^T는 A의 행과 열을 바꾼 것이고, $(A^T)^T$는 A^T의 행과 열을 바꾼 것이므로 $(A^T)^T = A$는 자명하다.

(3) $(cA)^T = cA^T$를 증명하기 위해 A를 $m \times n$ 행렬이라 하고 $B = cA$라고 한다. 그러면 $b_{ij} = ca_{ij}$이다. 전치행렬의 정의에 따라 어떤 $(i,\ j)$항에 대해

$$
\begin{aligned}
(cA)^T\text{의 } (i,\ j)\text{항} &= cA\text{의 } (j,\ i)\text{항} \\
&= b_{ji} \\
&= ca_{ji} \\
&= A^T\text{의 } c(i,\ j)
\end{aligned}
$$

■

예제 ❷−22 $A=\begin{bmatrix} 2 & 3 \\ -1 & 0 \end{bmatrix}$, $B=\begin{bmatrix} -1 & 5 \\ 2 & 1 \end{bmatrix}$일 때 $(AB)^T=B^T A^T$가 성립함을 살펴보자.

풀이 두 개의 식의 값을 구하여 그 값이 같음을 보인다.

$$(AB)^T=\left(\begin{bmatrix} 2 & 3 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} -1 & 5 \\ 2 & 1 \end{bmatrix}\right)^T=\left(\begin{bmatrix} 4 & 13 \\ 1 & -5 \end{bmatrix}\right)^T=\begin{bmatrix} 4 & 1 \\ 13 & -5 \end{bmatrix}$$

$$B^T\cdot A^T=\begin{bmatrix} -1 & 5 \\ 2 & 1 \end{bmatrix}^T\begin{bmatrix} 2 & 3 \\ -1 & 0 \end{bmatrix}^T=\begin{bmatrix} -1 & 2 \\ 5 & 1 \end{bmatrix}\begin{bmatrix} 2 & -1 \\ 3 & 0 \end{bmatrix}=\begin{bmatrix} 4 & 1 \\ 13 & -5 \end{bmatrix}$$

따라서 $(AB)^T=B^T A^T$가 성립함을 알 수 있으며 이 사실은 일반적으로 적용된다. ■

2.2 특수한 행렬

정의 ❷-14 어떤 정방행렬 $n \times n$ 행렬이 자신의 전치행렬과 똑같을 때, 즉 행렬 A가 $A = A^T$를 만족할 때 행렬 A를 **대칭행렬(symmetric matrix)**이라고 한다. 즉, $A = [a_{ij}]$에서 모든 i, j에 대해 $a_{ij} = a_{ji}$가 성립하는 경우이다.

예제 ❷-23 다음에 주어진 행렬들이 모두 대칭행렬임을 알아보자.

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 4 & 0 \\ 4 & 1 & 6 \\ 0 & 6 & 3 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$

풀이 행렬 A에서는 대각선을 중심으로 2와 2가 같고, 행렬 B에서는 대각선을 중심으로 4와 4, 0과 0, 6과 6이 같기 때문이다. 행렬 C의 경우 대각선을 중심으로 a_{12}와 a_{21}의 값이 2로써 같고, a_{31}과 a_{13}의 값이 3으로써 같으며 a_{32}와 a_{23}의 값이 5로써 같기 때문에 대칭행렬이 된다. ■

2.2 특수한 행렬

정의 ❷-15

$A=-A^T$를 만족하는 $n\times n$ 행렬을 교대행렬(skewed-symmetric matrix)이라고 한다. 즉, $A=[a_{ij}]$에서 모든 i, j에 대해 $a_{ij}=-a_{ji}$가 성립하는 경우이다.

예를 들면, $\begin{bmatrix} 0 & -2 \\ 2 & 0 \end{bmatrix}$과 $\begin{bmatrix} 0 & 4 & 0 \\ -4 & 0 & -6 \\ 0 & 6 & 0 \end{bmatrix}$은 둘 다 교대행렬이다.

예제 ❷-24 다음의 행렬들이 어떤 종류의 행렬인지를 알아보자.

$$A=\begin{bmatrix} 0 & 2 & 3 \\ -2 & 0 & -4 \\ -3 & 4 & 0 \end{bmatrix},\ B=\begin{bmatrix} a & h & g \\ h & b & f \\ g & f & c \end{bmatrix},\ C=\begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix}$$

풀이 행렬 A는 주대각선을 중심으로 -2와 2, 3과 -3, -4와 4가 부호가 다르면서 대칭을 이루기 때문에 교대행렬이다. 행렬 B는 주대각선을 중심으로 모든 항들이 대칭이므로 대칭행렬이지만, 행렬 C는 주대각선을 중심으로 부호만 다르기 때문에 교대행렬이다. ■

2.2 특수한 행렬

정의 ❷-16

주대각선 아래에 있는 모든 항들이 0인 $n \times n$ 행렬 A를 **상부삼각행렬(upper triangular matrix)**이라고 하며, 주대각선 위에 있는 모든 항들이 0인 $n \times n$ 행렬 A를 **하부삼각행렬(lower triangular matrix)**이라고 한다. 상부삼각행렬과 하부삼각행렬을 통칭하여 **삼각행렬(triangular matrix)**이라고 한다.

예를 들어, $\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$과 $\begin{bmatrix} 7 & 1 & -1 \\ 0 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix}$은 각각 2×2, 3×3 상부삼각행렬이고,

$\begin{bmatrix} 7 & 0 & 0 \\ 1 & -4 & 0 \\ -1 & 1 & -5 \end{bmatrix}$은 하부삼각행렬이다.

2.2 특수한 행렬

정리 ❷-6 상부삼각행렬의 전치는 하부삼각행렬이고, 하부삼각행렬의 전치는 상부삼각행렬이다.

예를 들어, 하부삼각행렬인 $\begin{bmatrix} 7 & 0 & 0 \\ 1 & -4 & 0 \\ -1 & 1 & -5 \end{bmatrix}$의 전치행렬은

$\begin{bmatrix} 7 & 1 & -1 \\ 0 & -4 & 1 \\ 0 & 0 & -5 \end{bmatrix}$이 되는데 이것은 상부삼각행렬이 된다.

상부삼각행렬과 하부삼각행렬은 통상 다음과 같이 간략히 표현될 수 있다.

상부삼각행렬은 $\begin{bmatrix} & * \\ 0 & \end{bmatrix}$로 나타내고 하부삼각행렬은 $\begin{bmatrix} & 0 \\ * & \end{bmatrix}$로 나타낸다.

여러 가지 행렬들의 예 2

전치행렬 $A=\begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 5 \end{bmatrix}$이면 $A^T=\begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 0 & 5 \end{bmatrix}$ 행과 열을 바꾸어 놓은 행렬

대칭행렬 $\begin{bmatrix} 2 & 4 & 0 \\ 4 & 1 & 6 \\ 0 & 6 & 3 \end{bmatrix}$ 대칭되는 행과 열을 바꾸어도 여전히 같은 행렬

교대행렬 $\begin{bmatrix} 2 & -4 & 0 \\ 4 & 1 & 6 \\ 0 & -6 & 3 \end{bmatrix}$ 대칭행렬 중 대칭되는 항들이 서로 −를 넣은 관계인 행렬

상부삼각행렬 $\begin{bmatrix} 7 & 1 & -1 \\ 0 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix}$ 주대각선 아래가 모두 0인 행렬

하부삼각행렬 $\begin{bmatrix} 7 & 0 & 0 \\ 1 & -4 & 0 \\ -1 & 1 & -5 \end{bmatrix}$ 주대각선 위가 모두 0인 행렬

2.3 행렬의 기본 연산과 사다리꼴

정의 ❷-17 어떤 행렬 A의 다음 세 가지 타입의 연산들을 **기본 행 연산(elementary row operation)**이라고 한다.

(1) Type Ⅰ : 어떤 2개의 행을 서로 바꾼다.

(2) Type Ⅱ : 어떤 행에다 0이 아닌 상수를 곱한다.

(3) Type Ⅲ : 어떤 행에다 상수를 곱한 후 다른 행에다 더한다.

정의 ❷-18 기본 행 연산을 필요에 따라 한 번 또는 여러 번 거친 것을 **행 동치(row equivalent)**라고 한다. 또한 $n \times n$ 항등행렬 I_n에서 한 번의 기본 행 연산을 거쳐 만들어지는 $n \times n$ 행렬을 **기본행렬(elementary matrix)**이라고 한다.

예제 ❷−25 3×3 항등행렬 A가 주어졌을 때, 다음의 기본 행 연산에 대응하는 행렬들을 각각 구해 보자.

$$A=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

풀이 (1) 1행의 성분과 2행의 성분을 서로 바꾼다. $[R_1 \leftrightarrow R_2]$

$$\begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix}$$

(2) 3행의 성분에다 (−2)를 곱한다. $[(-2)\times R_3 \rightarrow R_3]$

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

(3) 2행의 성분에다 (−3)을 곱하여 3행에 더한다. $[(-3)\times R_2 + R_3 \rightarrow R_3]$

$$\begin{bmatrix}1&0&0\\0&1&0\\0&-3&1\end{bmatrix}$$ ■

예제 ❷-26 다음 3개의 기본행렬이 항등행렬로부터 만들어지는 연산을 살펴보자.

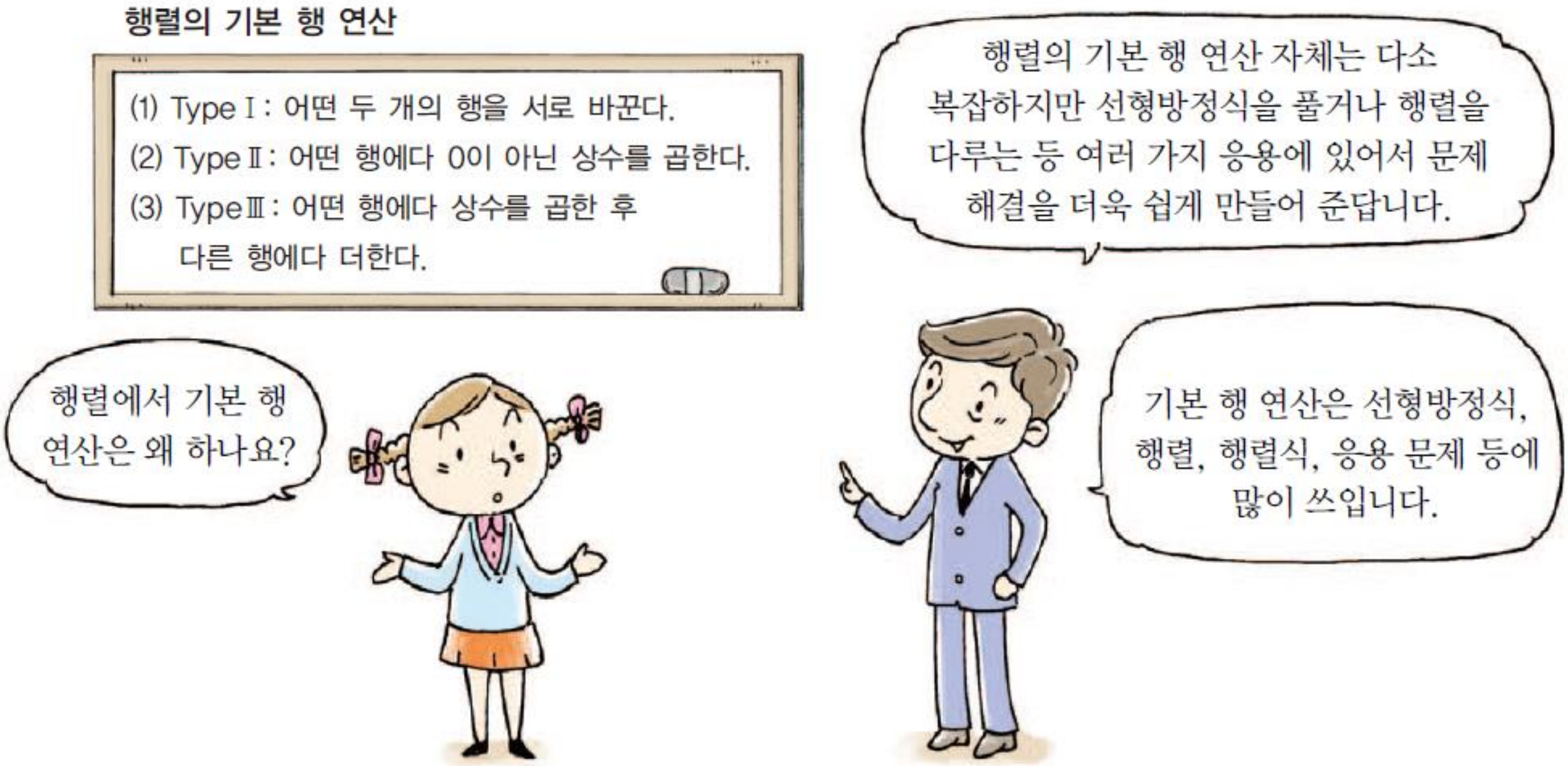
풀이 항등행렬로부터 기본행렬이 만들어진다.

$$\begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} \quad \leftarrow I_2\text{의 2행을 }(-2)\text{배하여 만들어진 행렬}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \leftarrow I_4\text{의 3행과 4행을 서로 바꾼 행렬}$$

$$\begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \leftarrow I_3\text{의 3행을 3배해서 1행에 더한 행렬}$$ ■

2.3 행렬의 기본 연산과 사다리꼴



2.3 행렬의 기본 연산과 사다리꼴

정의 ❷-19 행렬의 각 행에서 0이 아닌 가장 처음 나타나는 수 $a_{1j_1}, a_{2j_2}, \cdots, a_{rj_r}$를 행렬에서의 **피벗(pivot)**으로 삼을 수 있다.

정의 ❷-20 $m \times n$ 행렬 A가 기본 행 연산들을 거친 후 다음 3가지 조건을 만족시키면 **행 사다리꼴(row echelon form)**이라고 한다. 이것을 영어 약자로 **REF**로 표시하기도 한다.

(1) 0으로만 이루어진 행들은, 만약 있는 경우, 행렬의 아래쪽에 나타낸다.

(2) 모두가 0은 아닌 행의 가장 왼쪽에 가장 처음 나타나는 0이 아닌 수를 피벗으로 삼는다.

(3) 모두가 0은 아닌 연이은 두 행이 있으면 아래쪽 행의 피벗은 위쪽 행의 피벗보다 오른쪽에 있다.

2.3 행렬의 기본 연산과 사다리꼴

정의 ❷-21 $m \times n$ 행렬 A가 기본 행 연산들을 거친 후 행 사다리꼴(row echelon form)의 3가지 조건에다 다음의 4번째 조건까지를 만족시키면 기약 행 사다리꼴(reduced row echelon form)이라고 한다. 이것을 영어 약자로 RREF로 표시하기도 한다.

(4) 각 행의 피벗을 포함하는 열(column)에는 피벗 이외의 항들은 모두 0이다.

예제 ❷-27 다음 행렬이 모두 행 사다리꼴임을 알아보자.

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & 6 & 4 \\ 0 & 1 & 2 & -8 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 2 & 5 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 3 & 2 & 5 \\ 0 & 1 & 3 & 6 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

풀이 피벗의 아래 수들이 모두 0이므로 행 사다리꼴이다. ■

예제 ❷-28 다음 행렬이 모두 기약 행 사다리꼴임을 알아보자.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 2 & 5 \\ 0 & 1 & 3 & 6 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

풀이 피벗의 위와 아래의 수들이 모두 0이므로 기약 행 사다리꼴이다. ■

예제 ❷-29 다음의 어떤 행렬이 기약 행 사다리꼴이고, 어떤 행렬이 행 사다리꼴인지를 판단해 보자.

(1) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ (2) $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

(3) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ (4) $\begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 2 & 0 & 2 & 2 \\ 0 & 0 & 4 & 3 & 3 \\ 0 & 0 & 1 & 0 & 4 \end{bmatrix}$

풀이 (1), (2), (3)은 기약 행 사다리꼴이고, (4)는 어느 경우에도 해당되지 않는다. ■

2.3 행렬의 기본 연산과 사다리꼴

정의 ❷-22 기약 행 사다리꼴을 구하기 위한 기본 행 연산 방법은 다음과 같이 두 단계로 이루어진다. 전향단계(forward phase)에서는 피벗의 아랫부분이 0이 되도록 하고, 후향단계(backward phase)에서는 피벗의 윗부분까지 0이 되도록 행 연산을 실행한다. 전향단계까지만 연산 과정을 실행하면 행 사다리꼴을 구할 수 있는데, 이것을 가우스 소거법(Gauss elimination)이라고 한다. 한편 전향단계에다 후향단계까지 실행하면 가우스–조단 소거법(Gauss–Jordan elimination)이라고 한다.

예제 ❷-30 다음 행렬 A의 기약 행 사다리꼴을 구해 보자.

$$A=\begin{bmatrix} 1 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 \\ -3 & -3 & 1 & 1 \end{bmatrix}$$

풀이

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ \boxed{2} & 2 & 1 & 0 \\ -3 & -3 & 1 & 1 \end{bmatrix}$$

1 은 피벗 $\boxed{2}$ 는 소거될 항

$(-2)\times R_1+R_2\rightarrow R_2$

(1행에다 (−2)를 곱하여 2행에다 더한다.)

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \boxed{-3} & -3 & 1 & 1 \end{bmatrix}$$

$3\times R_1+R_3\rightarrow R_3$

(1행에다 3을 곱하여 3행에다 더한다.)

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \boxed{1} & 1 \end{bmatrix}$$

$(-1)\times R_2+R_3\rightarrow R_3$

(2행에다 (−1)을 곱하여 3행에다 더한다.)

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

그 결과 세 개 피벗의 위와 아래 항들이 모두 0이므로 기약 행 사다리꼴이다. ■

기본 행 연산에서 전향단계만 거치면 가우스 소거법이고, 후향단계까지 거치는 것을 가우스-조단 소거법이라고 합니다.
가우스-조단 소거법
• 전향단계
• 후향단계
가우스 소거법은 행 사다리꼴을 만들어 내고, 가우스-조단 소거법은 기약 행 사다리꼴을 만들어 낸다는 말씀이지요?
그렇지! 바로 그거예요. 행렬이나 선형방정식을 다룰 때 사용하는 매우 중요한 방법이지요.

예제 ❷-31 다음 행렬 A를 기약 행 사다리꼴로 바꿔 보자.

$$A=\begin{bmatrix}1 & 2 & 3 & 4\\ 5 & 6 & 7 & 8\\ 9 & 10 & 11 & 12\end{bmatrix}$$

풀이 먼저 주어진 행렬을 행 사다리꼴로 바꾼 후 계속해서 기약 행 사다리꼴로 바꾼다.

$$A=\begin{bmatrix}1 & 2 & 3 & 4\\ \boxed{5} & 6 & 7 & 8\\ 9 & 10 & 11 & 12\end{bmatrix} \quad (-5)\times R_1+R_2\to R_2$$

$$\begin{bmatrix}1 & 2 & 3 & 4\\ 0 & -4 & -8 & -12\\ \boxed{9} & 10 & 11 & 12\end{bmatrix} \quad (-9)\times R_1+R_3\to R_3$$

$$\begin{bmatrix}1 & 2 & 3 & 4\\ 0 & -4 & -8 & -12\\ 0 & \boxed{-8} & -16 & -24\end{bmatrix} \quad (-2)\times R_2+R_3\to R_3$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -4 & -8 & -12 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \left(-\frac{1}{4}\right)\times R_2 \rightarrow R_2$$

$$\begin{bmatrix} 1 & \boxed{2} & 3 & 4 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (-2)\times R_2 + R_1 \rightarrow R_1$$

$$\begin{bmatrix} 1 & 0 & -1 & -2 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

이 결과에서 알 수 있듯이 1행과 2행의 피벗 아래와 위는 모두 0이다. 따라서 이 결과는 기약 행 사다리꼴이다. ■

정의 ❷-23 주어진 행렬을 행 사다리꼴로 만들었을 때 행 전체가 0이 아닌 행의 개수는 수학적으로 중요한 의미를 가지는데, 이 수를 주어진 행렬의 **계수(rank)**라고 한다.

예제 ❷-32 다음은 피벗들이 동그라미로 그려진 행 사다리꼴이다. 이들 행렬의 계수는 무엇인가?

$$(1)\ \begin{bmatrix} 2 & 3 & 4 & 5 & 9 & 0 & 7 \\ 0 & 0 & 3 & 4 & 1 & 2 & 5 \\ 0 & 0 & 0 & 0 & 5 & 7 & 2 \\ 0 & 0 & 0 & 0 & 0 & 8 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (2)\ \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$(3)\ \begin{bmatrix} 2 & 3 & 2 & 0 & 4 & 5 & -6 \\ 0 & 0 & 0 & 1 & -3 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (4)\ \begin{bmatrix} 1 & 3 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 & -3 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

풀이 행렬의 계수는 피벗의 개수와 같으므로 4, 2, 3, 3이 된다. ■

MATLAB

예제 ❷-33

다음에 주어진 행렬 A의 계수를 구해 보자.

$$A=\begin{bmatrix}1 & 2 & -3 & 1 & 2\\ 2 & 4 & -4 & 6 & 10\\ 3 & 6 & -6 & 9 & 13\end{bmatrix}$$

풀이 먼저 a_{11} 아래의 항들을 0으로 만들기 위해 $a_{11}=1$을 피벗으로 사용한다.

$$\begin{bmatrix}1 & 2 & -3 & 1 & 2\\ \boxed{2} & 4 & -4 & 6 & 10\\ 3 & 6 & -6 & 9 & 13\end{bmatrix} \quad (-2)\times R_1+R_2\rightarrow R_2$$

$$\begin{bmatrix}1 & 2 & -3 & 1 & 2\\ 0 & 0 & 2 & 4 & 6\\ \boxed{3} & 6 & -6 & 9 & 13\end{bmatrix} \quad (-3)\times R_1+R_3\rightarrow R_3 \ \ (a_{11}=1\text{을 피벗으로 사용})$$

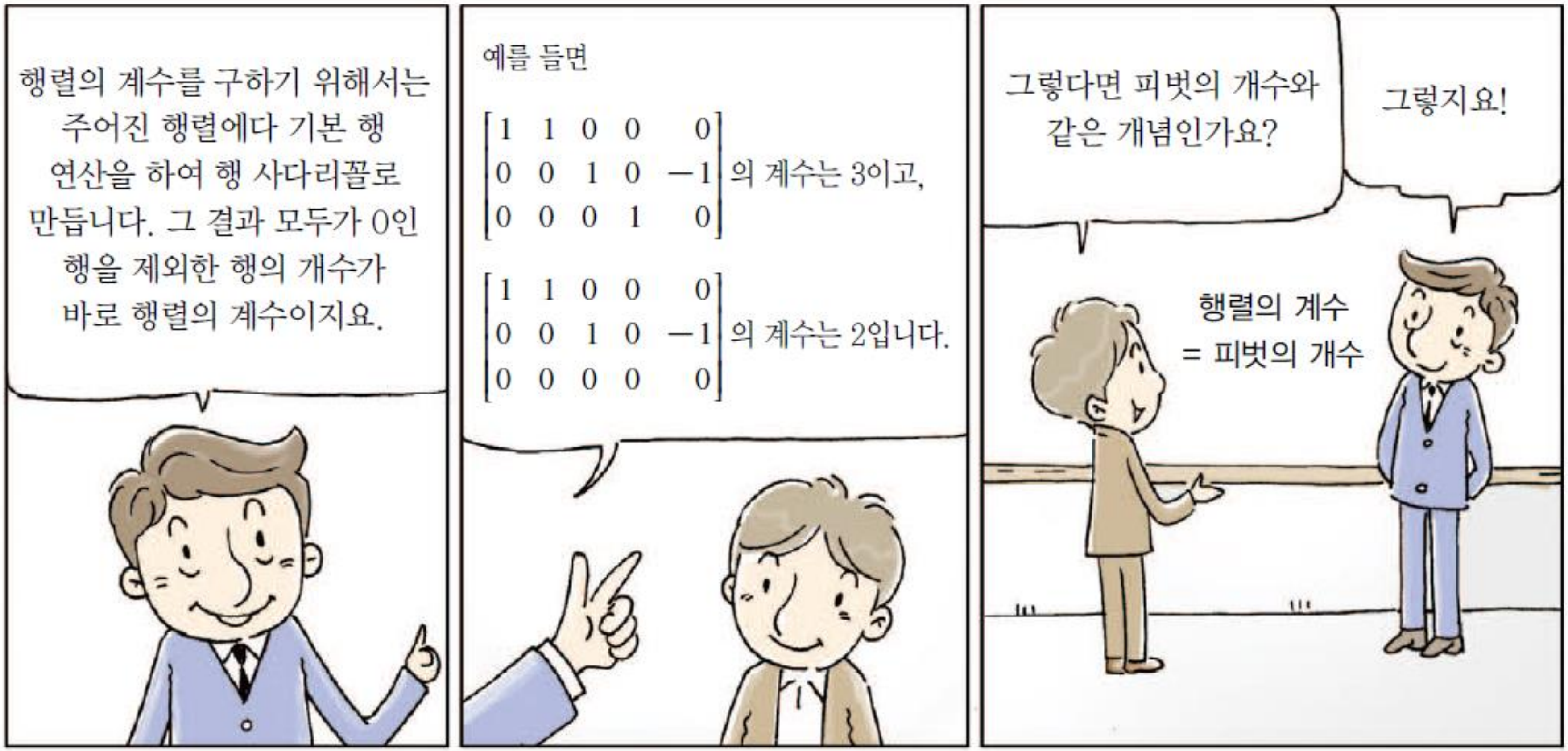
$$\begin{bmatrix}1 & 2 & -3 & 1 & 2\\ 0 & 0 & 2 & 4 & 6\\ 0 & 0 & \boxed{3} & 6 & 7\end{bmatrix} \quad \left(-\frac{3}{2}\right)\times R_2+R_3\rightarrow R_3 \ \ (a_{23}=2\text{를 피벗으로 사용})$$

그 결과 다음과 같은 행 사다리꼴이 만들어진다.

$$\begin{bmatrix} 1 & 2 & -3 & 1 & 2 \\ 0 & 0 & 2 & 4 & 6 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix}$$

따라서 주어진 행렬의 계수는 3이 된다. ■

2.3 행렬의 기본 연산과 사다리꼴



MATLAB

예제 ❷-34

다음 행렬 A의 계수를 구해 보자.

$$A=\begin{bmatrix}3 & -3 & 0\\ 1 & 4 & 6\\ 4 & 6 & -3\end{bmatrix}$$

풀이 주어진 행렬 A에 기본 행 연산을 필요에 따라 적용한다. 먼저 1행은 3으로 나누어지므로 일단 나누고 나서 그 수를 피벗으로 사용한다.

$$\begin{bmatrix}3 & -3 & 0\\ 1 & 4 & 6\\ 4 & 6 & -3\end{bmatrix}$$

$\left(\frac{1}{3}\right)\times R_1 \to R_1$

$$\begin{bmatrix}1 & -1 & 0\\ \boxed{1} & 4 & 6\\ 4 & 6 & -3\end{bmatrix}$$

$(-1)\times R_1 + R_2 \to R_2$

$$\begin{bmatrix} 1 & -1 & 0 \\ \boxed{0} & 5 & 6 \\ 4 & 6 & -3 \end{bmatrix}$$

$(-4) \times R_1 + R_3 \rightarrow R_3$

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 5 & 6 \\ 0 & \boxed{10} & -3 \end{bmatrix}$$

$(-2) \times R_2 + R_3 \rightarrow R_3$

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 5 & 6 \\ 0 & 0 & -15 \end{bmatrix}$$

그 결과 행의 각 항들이 모두가 0이 아닌 행의 개수가 3개이므로 행렬 A의 계수는 3이다. ■

예제 ❷-35 다음 행렬 A가 주어졌을 때 이 행렬의 계수를 구해 보자.

$$A = \begin{bmatrix} 1 & 1 & 2 & -1 \\ 1 & 2 & 1 & 0 \\ -1 & -4 & 1 & -2 \\ 1 & -2 & 5 & -4 \end{bmatrix}$$

풀이 A와 행 동치인 행렬을 만들기 위해 다음과 같은 여러 단계의 행 연산을 한다.

$$\begin{bmatrix} 1 & 1 & 2 & -1 \\ \boxed{1} & 2 & 1 & 0 \\ -1 & -4 & 1 & -2 \\ 1 & -2 & 5 & -4 \end{bmatrix} \qquad (-1)\times R_1 + R_2 \rightarrow R_2$$

$$\begin{bmatrix} 1 & 1 & 2 & -1 \\ 0 & 1 & -1 & 1 \\ \boxed{-1} & -4 & 1 & -2 \\ 1 & -2 & 5 & -4 \end{bmatrix}$$

$R_1 + R_3 \to R_3$

$$\begin{bmatrix} 1 & 1 & 2 & -1 \\ 0 & 1 & -1 & 1 \\ 0 & -3 & 3 & -3 \\ \boxed{1} & -2 & 5 & -4 \end{bmatrix}$$

$(-1) \times R_1 + R_4 \to R_4$

$$\begin{bmatrix} 1 & 1 & 2 & -1 \\ 0 & 1 & -1 & 1 \\ 0 & \boxed{-3} & 3 & -3 \\ 0 & -3 & 3 & -3 \end{bmatrix}$$

$3 \times R_2 + R_3 \to R_3$

$$\begin{bmatrix} 1 & 1 & 2 & -1 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & \boxed{-3} & 3 & -3 \end{bmatrix} \quad 3 \times R_2 + R_4 \rightarrow R_4$$

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

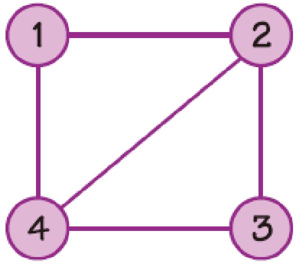
그 결과 행의 각 항들이 모두가 0이 아닌 행의 개수가 2개이므로 행렬 A의 계수는 2이다. ■

예제 ❷-36

행렬은 그래프의 표현이나 응용에도 폭넓게 활용될 수 있다. 어떤 점 i에서 점 j로의 연결이 있을 경우를 행렬로 나타낸 것을 인접행렬(adjacency matrix)이라고 하는데, 주어진 점의 개수가 n일 때 다음과 같이 $n \times n$ 행렬로 나타낸다. 만약 두 점 사이에 연결이 있을 경우에는 $a_{ij} = 1$이고, 그렇지 않은 경우에는 $a_{ij} = 0$이다.

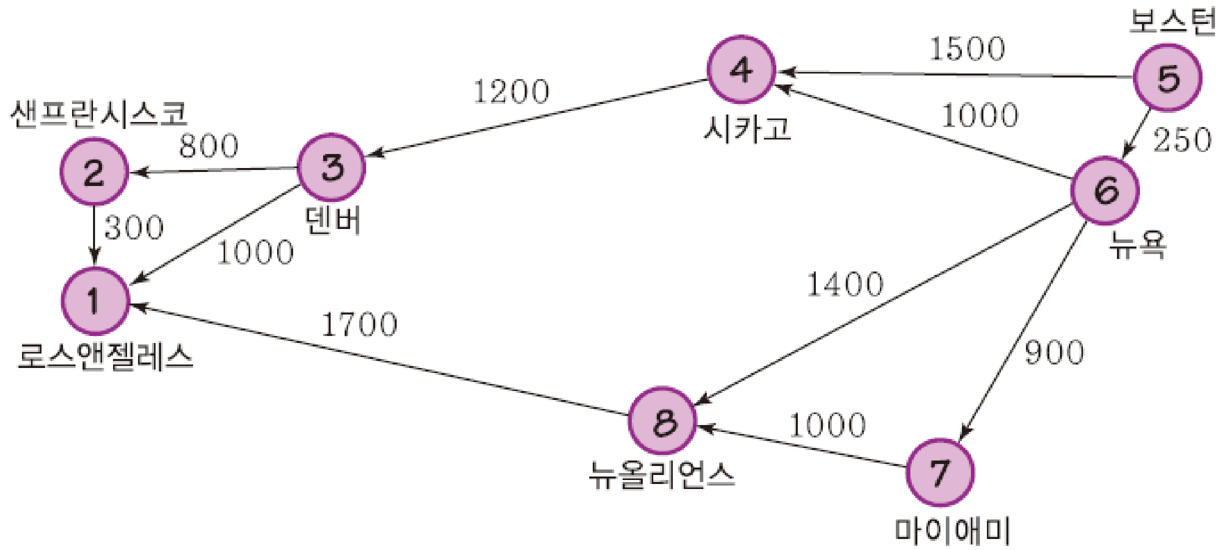
다음 〈그림 2.1〉의 그래프에 대응하는 인접행렬은 다음과 같다.

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$



〈그림 2.1〉 연결 그래프

이와 같은 개념을 확장하면 〈그림 2.2〉와 같은 최단 거리 경로(path), 통신 네트워크의 연결, 그래프 이론 등과 관련된 다양한 분야에 응용할 수 있다.



〈그림 2.2〉 최단 거리를 구하는 경로 문제

예제 ❷ −37

미국에 있는 어느 지방 도시의 패스트푸드 음식점에서는 햄버거(hamburger), 달걀(egg), 칩(chip), 삶은 콩(bean) 들을 결합한 세트메뉴를 제공한다. 그런데 주문하는 사람의 취향에 따라 A, B, C, D 4가지 형태의 세트메뉴를 제공한다.

A세트	–	칩 120g	삶은 콩 100g	햄버거 1개
B세트	달걀 1개	칩 250g	삶은 콩 150g	햄버거 1개
C세트	달걀 2개	칩 350g	삶은 콩 200g	햄버거 2개
D세트	달걀 1개	칩 200g	삶은 콩 150g	–

한 그룹의 단체 손님이 들어와서 A세트 1개, B세트 4개, C세트 2개, D세트 2개를 주문했다고 가정하면 주방에서는 얼마나 많은 음식 재료를 준비해야 할까? 그리고 고객 중 한 명은 C세트를 주문해 놓고 D세트로 주문을 변경할 수 있다고 가정한다. 이 경우에 주방에서는 원래보다 얼마나 적은 음식 재료를 준비해야 할까?

풀이 A, B, C, D 각 세트메뉴를 만드는 데 필요한 음식재료를 행렬로 나타내면 다음과 같다.

$$A=\begin{bmatrix} 0 \\ 120 \\ 100 \\ 1 \end{bmatrix},\quad B=\begin{bmatrix} 1 \\ 250 \\ 150 \\ 1 \end{bmatrix},\quad C=\begin{bmatrix} 2 \\ 350 \\ 200 \\ 2 \end{bmatrix},\quad D=\begin{bmatrix} 1 \\ 200 \\ 150 \\ 0 \end{bmatrix}$$

따라서 주방에서 필요한 음식 재료의 양은 다음과 같은 행렬로 표현할 수 있다.

$$A+4B+2C+2D=\begin{bmatrix} 10 \\ 2220 \\ 1400 \\ 9 \end{bmatrix}$$

어느 한 고객이 C세트에서 D세트로 주문을 변경할 경우에 달라지는 음식 재료의 행렬은 다음과 같이 변경될 것이다.

$$C-D=\begin{bmatrix} 2 \\ 350 \\ 200 \\ 2 \end{bmatrix}-\begin{bmatrix} 1 \\ 200 \\ 150 \\ 0 \end{bmatrix}=\begin{bmatrix} 1 \\ 150 \\ 50 \\ 2 \end{bmatrix}$$

비록 이런 예가 얼핏 보기에는 단순해 보일지는 몰라도 대량의 부품 공급을 필요로 하는 생산공정 등의 문제와 기본 원리는 같다. 그러므로 다양한 분야에 폭넓게 응용될 수 있다는 것을 보여 준다. ■

2.3 행렬의 기본 연산과 사다리꼴

예제 ❷-38 행렬은 함수의 변환을 위한 응용에 있어서 매우 편리한 계산을 제공할 수 있다. 예를 들면

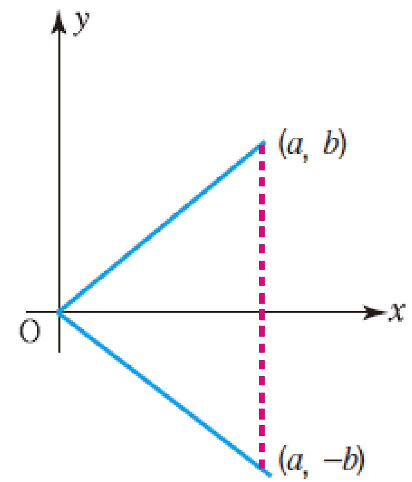
$$A=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix},\quad x=\begin{bmatrix}a\\ b\end{bmatrix} \text{ 일 때}$$

$y=Ax$를 구해 보자.

풀이

$$y=Ax=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}a\\ b\end{bmatrix}=\begin{bmatrix}a\\ -b\end{bmatrix}$$

이므로 a는 그대로 두고 b는 $-b$에 대응시킨다. 이것은 〈그림 2.3〉과 같이 주어진 좌표 $(a,\ b)$를 x축에 반사시킨 $(a,\ -b)$로 대응시키는 역할을 한다.

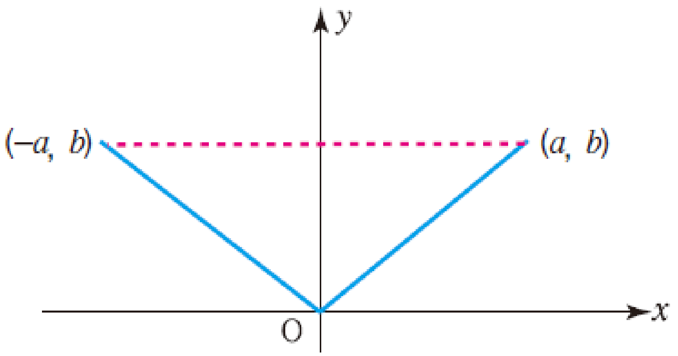


〈그림 2.3〉 행렬의 함수 변환에의 적용

만약 $A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$ 이라면

$y = Ax = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -a \\ b \end{bmatrix}$ 가 되어

〈그림 2.4〉와 같이 y축 반사의 역할을 한다.

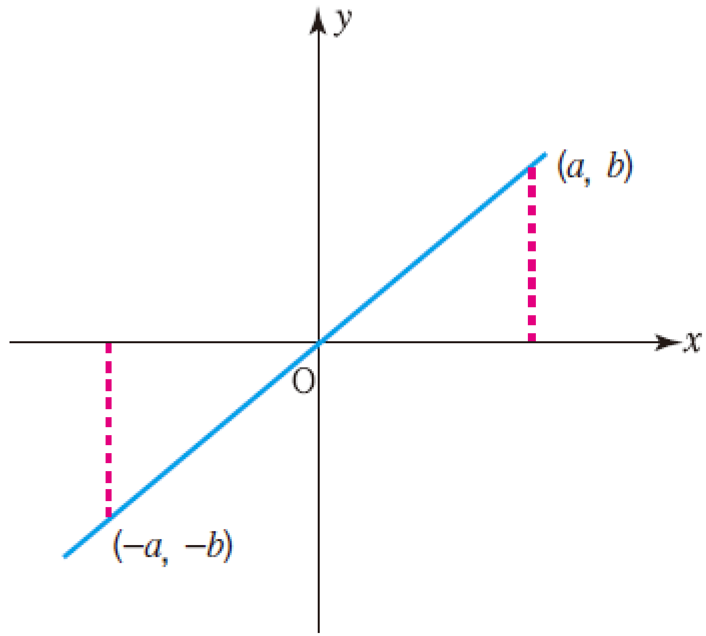


〈그림 2.4〉 행렬의 y축 반사 변환

또한 $A=\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}$이라면

$$y=Ax=\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}a\\ b\end{bmatrix}=\begin{bmatrix}-a\\ -b\end{bmatrix}$$ 가 되어

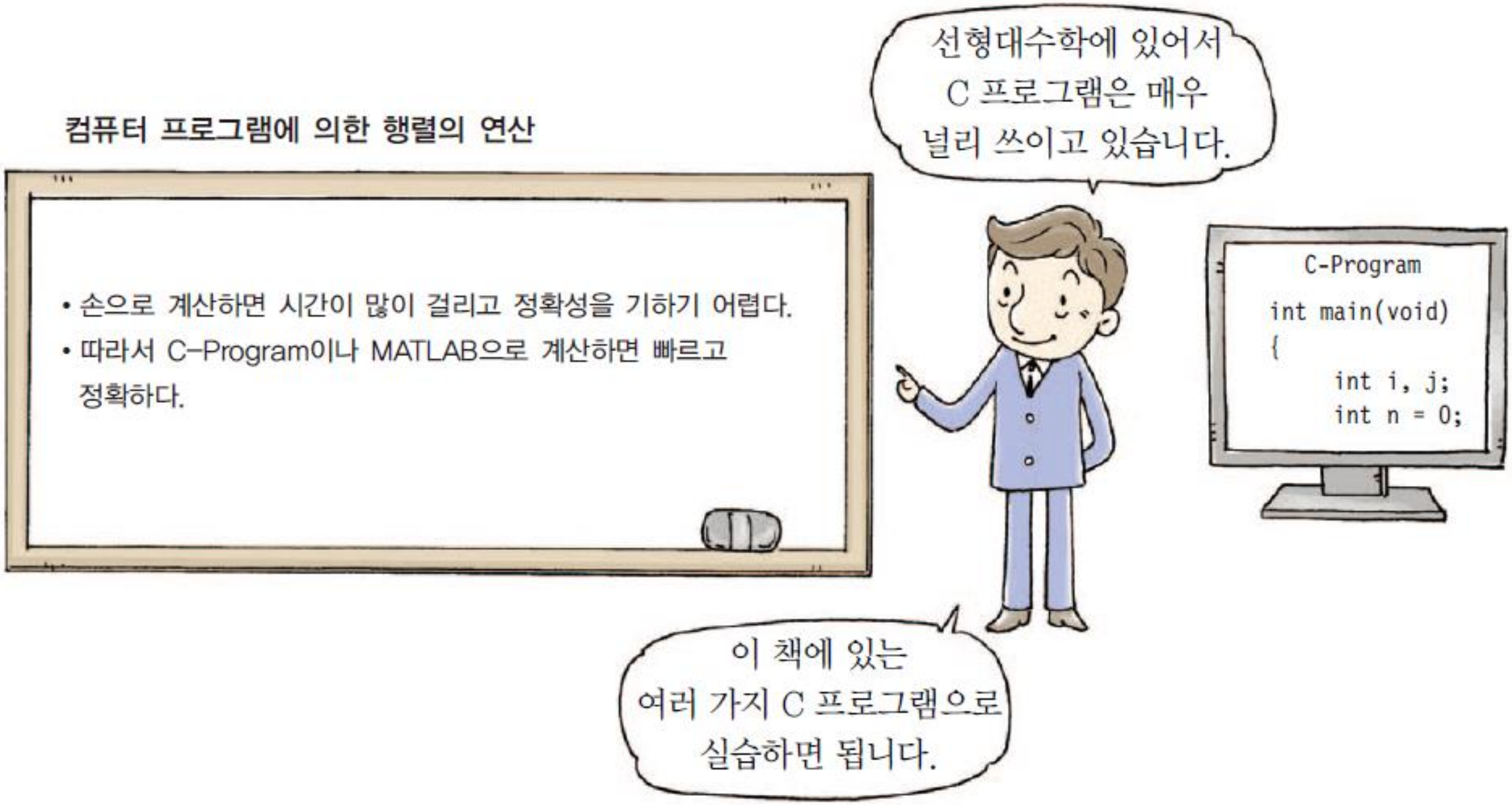
〈그림 2.5〉와 같이 행렬의 원점 대칭의 역할을 한다.



〈그림 2.5〉 행렬의 원점 대칭 변환

2.4 컴퓨터 프로그램에 의한 연산

컴퓨터 프로그램에 의한 행렬의 연산



2.4 컴퓨터 프로그램에 의한 연산

```
#include <stdio.h>

#define MAX 30                          // 배열의 임시 최대값 선언

int main(void)
{
   int i, j;                            // 루프를 수행하기 위한 변수 선언
int n = 0;                              // 행렬의 행 변수 값(n x m 행렬)
   int m = 0;                           // 행렬의 열 변수 값
   int vMatrixA[MAX][MAX] = {0,};       // 행렬 A 배열 선언
   int vMatrixB[MAX][MAX] = {0,};       // 행렬 B 배열 선언
   int vResult[MAX][MAX] = {0,};        // 행렬 덧셈 결과 값 배열 선언

   printf(" ******************************************************\n");
   printf(" **                                                  **\n");
   printf(" **                행렬의 합 계산 프로그램               **\n");
   printf(" **                                                  **\n");
   printf(" *****************************************************\n\n");
```

```
// 행렬의 행과 열 개수 입력(행렬의 덧셈은 두 행렬이 같아야 하기 때문에 한 번
   만 입력한다)
printf(" 덧셈하려는 행렬의 크기를 입력하세요\n");
printf(" 행렬의 행 크기 입력: ");
scanf("%d", &n);
printf(" 행렬의 열 크기 입력: ");
scanf("%d", &m);

// 첫 번째 행렬 값 입력(A 행렬)
printf("\n");
printf(" 첫번째 행렬의 값을 입력하세요. \n");

// 입력받은 값을 A 행렬 배열에 넣는다.
for(i = 0; i < n; i++)
{
    for(j = 0; j < m; j++)
    {
        printf(" %d X %d 행렬의 값을 입력하세요: ", i+1, j+1);
        scanf("%d", &vMatrixA[i][j]);
    }
}
```

```
// 두 번째 행렬 값 입력(B 행렬)
printf("\n");
printf(" 두번째 행렬의 값을 입력하세요. \n");

// 입력받은 값을 B 행렬 배열에 삽입한다
for(i = 0; i < n; i++)
{
    for(j = 0; j < m; j++)
    {
        printf(" %d X %d 행렬의 값을 입력하세요: ", i+1, j+1);
        scanf("%d", &vMatrixB[i][j]);
    }
}

// 행렬 A + B 연산
// 행렬의 덧셈은 두 행렬의 크기가 같아야 하므로 행렬 크기에 변화가 없다(입력
   받은 n과 m의 크기 즉 n x m 형태의 행렬 결과 값)
```

```
for(i = 0; i < n; i++)
{
    for(j = 0; j < m; j++)
    {
        vResult[i][j] += vMatrixA[i][j] + vMatrixB[i][j];
    }
}

// A + B의 결과 값을 출력한다.
printf("\n");
printf(" 두 행렬 덧셈의 결과값\n");
```

```
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < m; j++)
        {
            printf("%4d ", vResult[i][j]);
        }
        printf("\n");
    }
    printf("\n");

    return 0;
}
```

실습 ❷-1

예제 ②-2 C program, MATLAB

$A=\begin{bmatrix}3 & -1\\ 2 & 4\end{bmatrix}$, $B=\begin{bmatrix}2 & 1\\ -3 & 2\end{bmatrix}$일 때 $A+B$를 구해 보자.

풀이 $$A+B=\begin{bmatrix}3+2 & -1+1\\ 2+(-3) & 4+2\end{bmatrix}=\begin{bmatrix}5 & 0\\ -1 & 6\end{bmatrix}$$

이 된다. ■

```
C:\WINDOWS\system32\cmd.exe
******************************************************
**                                                  **
**               행렬의 합 계산 프로그램            **
**                                                  **
******************************************************

덧셈하려는 행렬의 크기를 입력하세요
행렬의 행 크기 입력 : 2
행렬의 열 크기 입력 : 2

첫번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 3
1 X 2 행렬의 값을 입력하세요 : -1
2 X 1 행렬의 값을 입력하세요 : 2
2 X 2 행렬의 값을 입력하세요 : 4

두번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 2
1 X 2 행렬의 값을 입력하세요 : 1
2 X 1 행렬의 값을 입력하세요 : -3
2 X 2 행렬의 값을 입력하세요 : 2

두 행렬 덧셈의 결과값
  5    0
 -1    6

계속하려면 아무 키나 누르십시오 . . .
```

예제 ②-3 C program, MATLAB

실습 ❷-2

다음 두 행렬 A, B의 합을 구해 보자.

$$A=\begin{bmatrix}3 & -1 & 3\\ 0 & 4 & 6\\ 2 & 7 & -5\end{bmatrix},\quad B=\begin{bmatrix}0 & -2 & -4\\ 1 & 6 & -2\\ 1 & -1 & 2\end{bmatrix}$$

풀이

$$A+B=\begin{bmatrix}3+0 & -1+(-2) & 3+(-4)\\ 0+1 & 4+6 & 6+(-2)\\ 2+1 & 7+(-1) & (-5)+2\end{bmatrix}=\begin{bmatrix}3 & -3 & -1\\ 1 & 10 & 4\\ 3 & 6 & -3\end{bmatrix}$$

컴퓨터 프로그램에 의한 연산

```
C:\WINDOWS\system32\cmd.exe
******************************************************
**                                                  **
**              행렬의 합 계산 프로그램             **
**                                                  **
******************************************************

덧셈하려는 행렬의 크기를 입력하세요
행렬의 행 크기 입력 : 3
행렬의 열 크기 입력 : 3

첫번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 3
1 X 2 행렬의 값을 입력하세요 : -1
1 X 3 행렬의 값을 입력하세요 : 3
2 X 1 행렬의 값을 입력하세요 : 0
2 X 2 행렬의 값을 입력하세요 : 4
2 X 3 행렬의 값을 입력하세요 : 6
3 X 1 행렬의 값을 입력하세요 : 2
3 X 2 행렬의 값을 입력하세요 : 7
3 X 3 행렬의 값을 입력하세요 : -5

두번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 0
1 X 2 행렬의 값을 입력하세요 : -2
1 X 3 행렬의 값을 입력하세요 : -4
2 X 1 행렬의 값을 입력하세요 : 1
2 X 2 행렬의 값을 입력하세요 : 6
2 X 3 행렬의 값을 입력하세요 : -2
3 X 1 행렬의 값을 입력하세요 : 1
3 X 2 행렬의 값을 입력하세요 : -1
3 X 3 행렬의 값을 입력하세요 : 2

두 행렬 덧셈의 결과값
   3   -3   -1
   1   10    4
   3    6   -3

계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

#define MAX 30                          // 배열의 임시 최대값 선언

int main(void)
{
   int i, j, k;                         // 루프를 수행하기 위한 변수 선언
   int An = 0;                          // A 행렬의 행 변수 값
   int Am = 0;                          // A 행렬의 열 변수 값
   int Bn = 0;                          // B 행렬의 행 변수 값
   int Bm = 0;                          // B 행렬의 열 변수 값
   int vMatrixA[MAX][MAX] = {0,};       // 행렬 A 배열 선언
   int vMatrixB[MAX][MAX] = {0,};       // 행렬 B 배열 선언
   int vResult[MAX][MAX] = {0,};        // 행렬 곱 결과 값 배열 선언

   printf(" ******************************************************\n");
   printf(" **                                                  **\n");
   printf(" **                행렬의 곱 계산 프로그램             **\n");
   printf(" **                                                  **\n");
   printf(" *****************************************************\n\n");
```

```
// 첫 번째 행렬을 입력한다(A 행렬)
printf(" 첫번째 행렬을 입력하세요\n");
printf(" 첫번째 행렬의 행 크기 입력: ");
scanf("%d", &An);
printf(" 첫번째 행렬의 열 크기 입력: ");
scanf("%d", &Am);

printf("\n");
printf(" 첫번째 행렬의 값을 입력하세요. \n");

// 입력받은 값을 A 행렬 배열에 넣는다.
for(i = 0; i < An; i++)
{
```

```
        for(j = 0; j < Am; j++)
        {
            printf(" %d X %d 행렬의 값을 입력하세요: ", i+1, j+1);
            scanf("%d", &vMatrixA[i][j]);
        }
    }

    // 두 번째 행렬을 입력한다(B 행렬)
    printf("\n");
    printf(" 두번째 행렬을 입력하세요\n");
    printf(" 두번째 행렬의 행 크기 입력: ");
    scanf("%d", &Bn);
    printf(" 두번째 행렬의 열 크기 입력: ");
    scanf("%d", &Bm);

    printf("\n");
    printf(" 두번째 행렬의 값을 입력하세요. \n");
```

```
// 입력받은 값을 B 행렬 배열에 삽입한다.
for(i = 0; i < Bn; i++)
{
    for(j = 0; j < Bm; j++)
    {
        printf(" %d X %d 행렬의 값을 입력하세요: ", i+1, j+1);
        scanf("%d", &vMatrixB[i][j]);
    }
}

// 행렬의 곱 A X B 연산
// 행렬 곱 연산의 결과는 행렬의 형태가 (A 행렬의 행 x B 행렬의 열)의 형태로
   나타난다.(An x Bm의 행렬 형태)
for(i = 0; i < An; i++)
{
    for(j = 0; j < Bm; j++)
    {
        vResult[i][j] = 0;            // 결과 값 행렬을 초기화한다.
```

```
            for(k = 0; k < Am; k++)
            {
                vResult[i][j] += vMatrixA[i][k] * vMatrixB[k][j];
            }
        }
    }

    // A X B의 결과 값을 출력한다.
    printf("\n");
    printf(" 두 행렬 곱의 결과값\n");

    for(i = 0; i < An; i++)
    {
        for(j = 0; j < Bm; j++)
        {
            printf(" %4d ", vResult[i][j]);
        }
        printf("\n");
    }
    printf("\n");

    return 0;
}
```

실습 ❷-3

예제 ②-9 C program, MATLAB

행렬 A와 행렬 B가 다음과 같이 주어졌을 때, 행렬의 곱 AB의 두 번째 행의 값을 구해 보자.

$$A=\begin{bmatrix} 2 & 4 & -1 \\ -1 & 3 & 3 \\ 4 & -2 & 1 \\ -3 & 0 & 2 \end{bmatrix},\quad B=\begin{bmatrix} 4 & -2 \\ -2 & 1 \\ 3 & -1 \end{bmatrix}$$

풀이 $$AB=\begin{bmatrix} -3 & 1 \\ -1 & 2 \\ 23 & -11 \\ -6 & 4 \end{bmatrix}$$ ■

2.4 컴퓨터 프로그램에 의한 연산

```
C:₩WINDOWS₩system32₩cmd.exe
******************************************************
**                                                  **
**              행렬의 곱 계산 프로그램              **
**                                                  **
******************************************************

첫번째 행렬을 입력하세요
첫번째 행렬의 행 크기 입력 : 4
첫번째 행렬의 열 크기 입력 : 3

첫번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 2
1 X 2 행렬의 값을 입력하세요 : 4
1 X 3 행렬의 값을 입력하세요 : -1
2 X 1 행렬의 값을 입력하세요 : -1
2 X 2 행렬의 값을 입력하세요 : 3
2 X 3 행렬의 값을 입력하세요 : 3
3 X 1 행렬의 값을 입력하세요 : 4
3 X 2 행렬의 값을 입력하세요 : -2
3 X 3 행렬의 값을 입력하세요 : 1
4 X 1 행렬의 값을 입력하세요 : -3
4 X 2 행렬의 값을 입력하세요 : 0
4 X 3 행렬의 값을 입력하세요 : 2

두번째 행렬을 입력하세요
두번째 행렬의 행 크기 입력 : 3
두번째 행렬의 열 크기 입력 : 2

두번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 4
1 X 2 행렬의 값을 입력하세요 : -2
2 X 1 행렬의 값을 입력하세요 : -2
2 X 2 행렬의 값을 입력하세요 : 1
3 X 1 행렬의 값을 입력하세요 : 3
3 X 2 행렬의 값을 입력하세요 : -1

두 행렬 곱의 결과값
  -3     1
  -1     2
  23   -11
  -6     4

계속하려면 아무 키나 누르십시오 . . .
```

2.4 컴퓨터 프로그램에 의한 연산

실습 ❷-4

예제 ②-10 C program, MATLAB

행렬 A, B, C가 다음과 같이 주어졌을 때, AB와 AC를 구해 보자.

$$A=\begin{bmatrix}1 & 1 & 0\\2 & 0 & 1\end{bmatrix},\quad B=\begin{bmatrix}2 & 0\\0 & 1\\1 & 3\end{bmatrix},\quad C=\begin{bmatrix}1 & -2\\-1 & 2\\-2 & 4\end{bmatrix}$$

풀이 (1) $AB=\begin{bmatrix}1 & 1 & 0\\2 & 0 & 1\end{bmatrix}\begin{bmatrix}2 & 0\\0 & 1\\1 & 3\end{bmatrix}$

$$=\begin{bmatrix}(1)(2)+(1)(0)+(0)(1) & (1)(0)+(1)(1)+(0)(3)\\(2)(2)+(0)(0)+(1)(1) & (2)(0)+(0)(1)+(1)(3)\end{bmatrix}=\begin{bmatrix}2 & 1\\5 & 3\end{bmatrix}\quad \blacksquare$$

여기서는 AB의 값만 구해 본다.

2.4 컴퓨터 프로그램에 의한 연산

C:\WINDOWS\system32\cmd.exe

```
*******************************************************
**                                                   **
**                행렬의 곱 계산 프로그램                **
**                                                   **
*******************************************************

첫번째 행렬을 입력하세요
첫번째 행렬의 행 크기 입력 : 2
첫번째 행렬의 열 크기 입력 : 3

첫번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 1
1 X 2 행렬의 값을 입력하세요 : 1
1 X 3 행렬의 값을 입력하세요 : 0
2 X 1 행렬의 값을 입력하세요 : 2
2 X 2 행렬의 값을 입력하세요 : 0
2 X 3 행렬의 값을 입력하세요 : 1

두번째 행렬을 입력하세요
두번째 행렬의 행 크기 입력 : 3
두번째 행렬의 열 크기 입력 : 2

두번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 2
1 X 2 행렬의 값을 입력하세요 : 0
2 X 1 행렬의 값을 입력하세요 : 0
2 X 2 행렬의 값을 입력하세요 : 1
3 X 1 행렬의 값을 입력하세요 : 1
3 X 2 행렬의 값을 입력하세요 : 3

두 행렬 곱의 결과값
   2      1
   5      3

계속하려면 아무 키나 누르십시오 . . .
```

2.4 컴퓨터 프로그램에 의한 연산

실습 ❷-5

예제 ②-12 C program, MATLAB

다음의 A, B 두 행렬이 주어졌을 때, 이 행렬들의 곱셈에서는 교환법칙이 성립하지 않음을 살펴보자. 즉, $AB \neq BA$임을 보인다.

$$A = \begin{bmatrix} 5 & 1 \\ 3 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 4 & 3 \end{bmatrix}$$

풀이 AB와 BA의 값이 다름을 보인다.

$$AB = \begin{bmatrix} 5 & 1 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 14 & 3 \\ -2 & -6 \end{bmatrix}$$

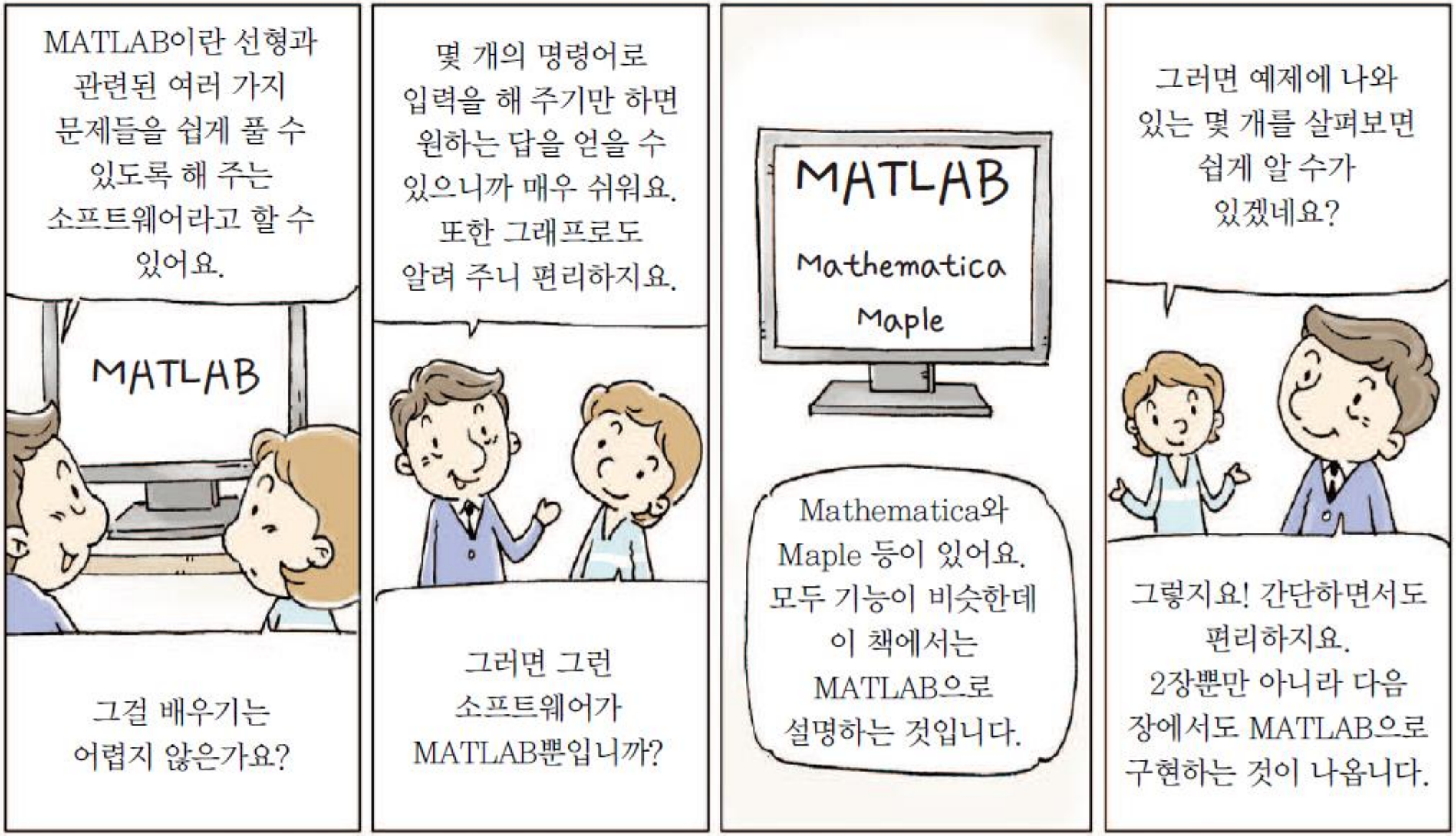
$$BA = \begin{bmatrix} 2 & 0 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 5 & 1 \\ 3 & -2 \end{bmatrix} = \begin{bmatrix} 10 & 2 \\ 29 & -2 \end{bmatrix} \quad ■$$

여기서는 AB의 값만 구해 본다.

2.4 컴퓨터 프로그램에 의한 연산

```
C:\WINDOWS\system32\cmd.exe
******************************************************
**                                                  **
**              행렬의 곱 계산 프로그램             **
**                                                  **
******************************************************

첫번째 행렬을 입력하세요
첫번째 행렬의 행 크기 입력 : 2
첫번째 행렬의 열 크기 입력 : 2

첫번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 5
1 X 2 행렬의 값을 입력하세요 : 1
2 X 1 행렬의 값을 입력하세요 : 3
2 X 2 행렬의 값을 입력하세요 : -2

두번째 행렬을 입력하세요
두번째 행렬의 행 크기 입력 : 2
두번째 행렬의 열 크기 입력 : 2

두번째 행렬의 값을 입력하세요.
1 X 1 행렬의 값을 입력하세요 : 2
1 X 2 행렬의 값을 입력하세요 : 0
2 X 1 행렬의 값을 입력하세요 : 4
2 X 2 행렬의 값을 입력하세요 : 3

두 행렬 곱의 결과값
  14      3
  -2     -6

계속하려면 아무 키나 누르십시오 . . .
```

2.4 컴퓨터 프로그램에 의한 연산



2.4 컴퓨터 프로그램에 의한 연산

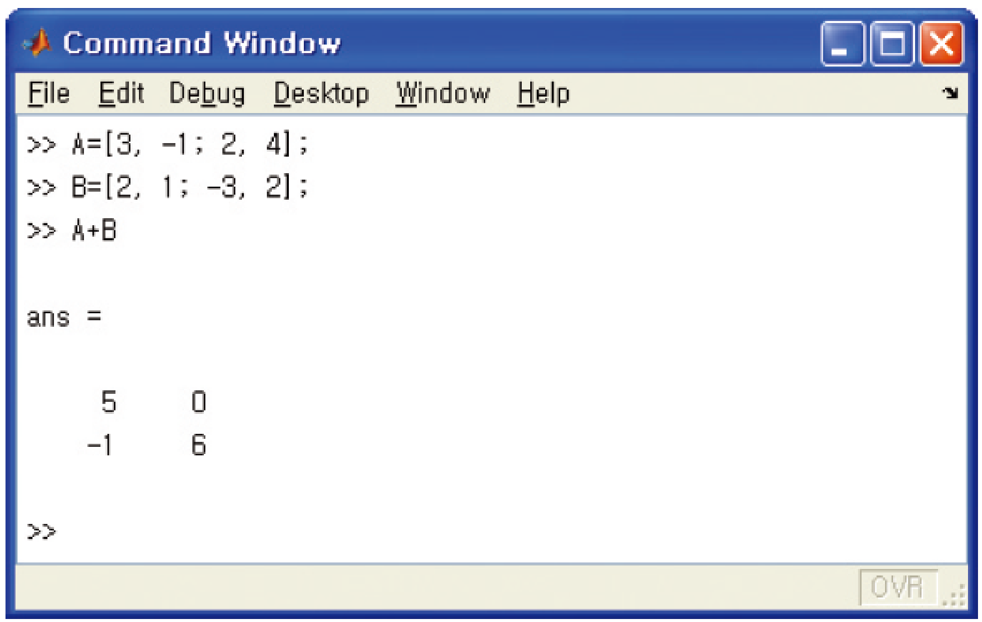
실습 ❷-6

예제 ②-2 C program, MATLAB

$A=\begin{bmatrix} 3 & -1 \\ 2 & 4 \end{bmatrix}$, $B=\begin{bmatrix} 2 & 1 \\ -3 & 2 \end{bmatrix}$일 때 $A+B$를 구해 보자.

풀이 $$A+B=\begin{bmatrix} 3+2 & (-1)+1 \\ 2+(-3) & 4+2 \end{bmatrix}=\begin{bmatrix} 5 & 0 \\ -1 & 6 \end{bmatrix}$$

이 된다. ■



2.4 컴퓨터 프로그램에 의한 연산

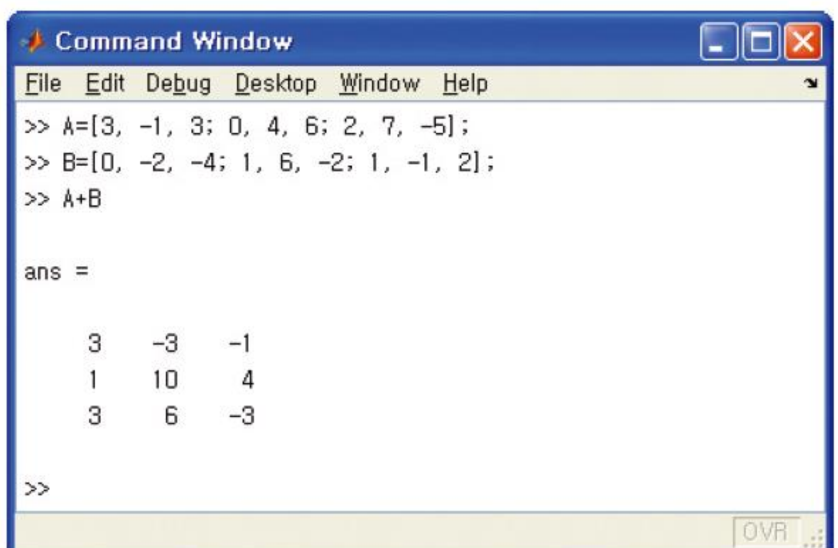
예제 ❷-3 C program, MATLAB

실습 ❷-7

다음 두 행렬 A, B의 합을 구해 보자.

$$A=\begin{bmatrix}3&-1&3\\0&4&6\\2&7&-5\end{bmatrix},\quad B=\begin{bmatrix}0&-2&-4\\1&6&-2\\1&-1&2\end{bmatrix}$$

풀이 $A+B=\begin{bmatrix}3+0&(-1)+(-2)&3+(-4)\\0+1&4+6&6+(-2)\\2+1&7+(-1)&(-5)+2\end{bmatrix}=\begin{bmatrix}3&-3&-1\\1&10&4\\3&6&-3\end{bmatrix}$ ■



2.4 컴퓨터 프로그램에 의한 연산

실습 ❷-8

예제 ②-9 C program, MATLAB

행렬 A와 행렬 B가 다음과 같이 주어졌을 때, 행렬의 곱 AB의 두 번째 행의 값을 구해 보자.

$$A=\begin{bmatrix} 2 & 4 & -1 \\ -1 & 3 & 3 \\ 4 & -2 & 1 \\ -3 & 0 & 2 \end{bmatrix},\quad B=\begin{bmatrix} 4 & -2 \\ -2 & 1 \\ 3 & -1 \end{bmatrix}$$

풀이 AB의 값을 구하면

$$AB=\begin{bmatrix} -3 & 1 \\ -1 & 2 \\ 23 & -11 \\ -6 & 4 \end{bmatrix}$$

가 된다. ■

2.4 컴퓨터 프로그램에 의한 연산

Command Window

File Edit Debug Desktop Window Help

```
>> A=[2, 4, -1; -1, 3, 3; 4, -2, 1; -3, 0, 2];
>> B=[4, -2; -2, 1; 3, -1];
>> A*B

ans =

    -3     1
    -1     2
    23   -11
    -6     4

>>
```

OVR

실습 ❷-9

예제 ②-10 C program, MATLAB

행렬 A, B, C가 다음과 같이 주어졌을 때, AB와 AC를 구해 보자.

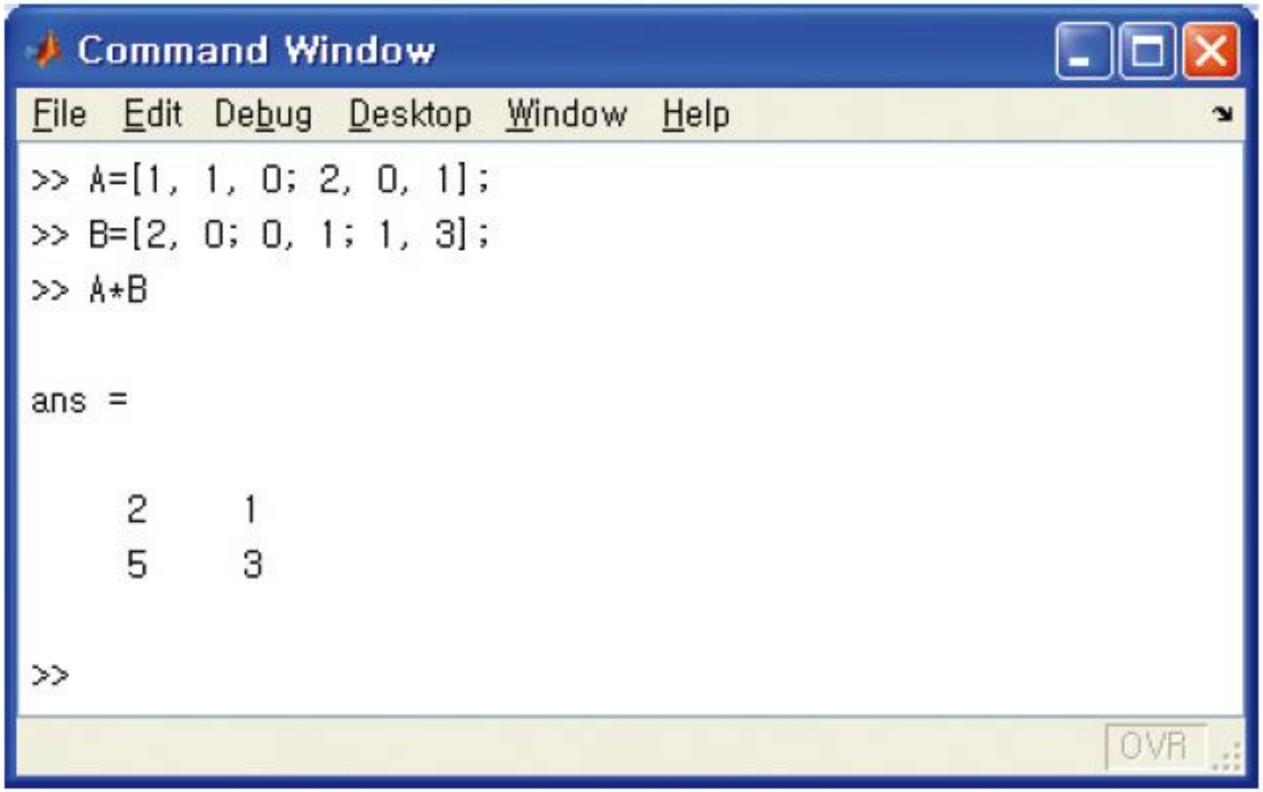
$$A=\begin{bmatrix}1 & 1 & 0\\ 2 & 0 & 1\end{bmatrix},\quad B=\begin{bmatrix}2 & 0\\ 0 & 1\\ 1 & 3\end{bmatrix},\quad C=\begin{bmatrix}1 & -2\\ -1 & 2\\ -2 & 4\end{bmatrix}$$

풀이 (1) $AB=\begin{bmatrix}1 & 1 & 0\\ 2 & 0 & 1\end{bmatrix}\begin{bmatrix}2 & 0\\ 0 & 1\\ 1 & 3\end{bmatrix}$

$$=\begin{bmatrix}(1)(2)+(1)(0)+(0)(1) & (1)(0)+(1)(1)+(0)(3)\\ (2)(2)+(0)(0)+(1)(1) & (2)(0)+(0)(1)+(1)(3)\end{bmatrix}=\begin{bmatrix}2 & 1\\ 5 & 3\end{bmatrix}$$ ■

여기서는 AB의 값만 구해 본다.

컴퓨터 프로그램에 의한 연산



실습 ❷-10

예제 ②-12 C program, MATLAB

다음의 A, B 두 행렬이 주어졌을 때, 이 행렬들의 곱셈에서는 교환법칙이 성립하지 않음을 살펴보자. 즉, $AB \neq BA$임을 보인다.

$$A = \begin{bmatrix} 5 & 1 \\ 3 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 4 & 3 \end{bmatrix}$$

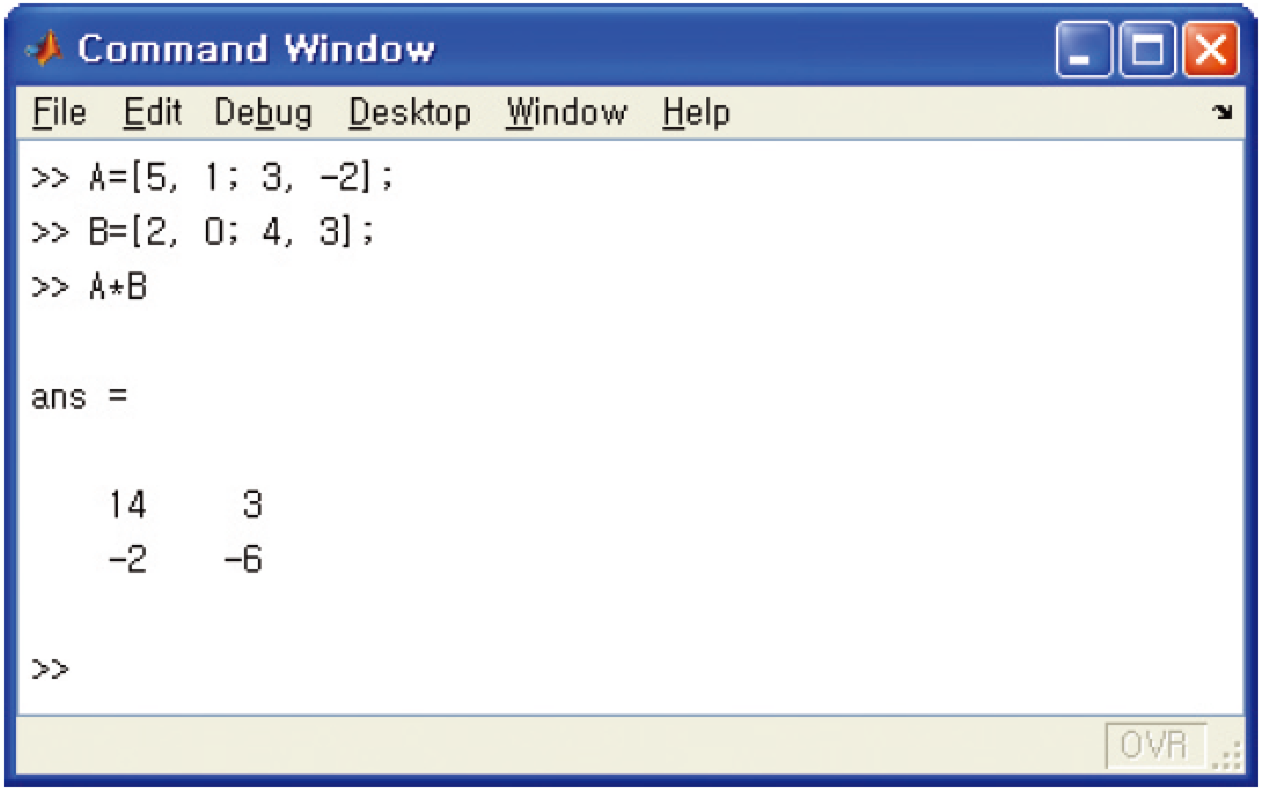
풀이 AB와 BA의 값이 다르다는 것을 보인다.

$$AB = \begin{bmatrix} 5 & 1 \\ 3 & -2 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 14 & 3 \\ -2 & -6 \end{bmatrix}$$

$$BA = \begin{bmatrix} 2 & 0 \\ 4 & 3 \end{bmatrix}\begin{bmatrix} 5 & 1 \\ 3 & -2 \end{bmatrix} = \begin{bmatrix} 10 & 2 \\ 29 & -2 \end{bmatrix} \quad ■$$

여기서는 AB의 값만 구해 본다.

컴퓨터 프로그램에 의한 연산



실습 ❷-11

예제 ②-19 MATLAB

다음 행렬 A, B, C에서 대각항과 대각합을 구해 보자.

$$(1)\ A=\begin{bmatrix}1 & 3 & 6\\ 2 & -5 & 8\\ 4 & -2 & 9\end{bmatrix} \qquad (2)\ B=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix} \qquad (3)\ C=\begin{bmatrix}1 & 2 & -3\\ 4 & -5 & 6\end{bmatrix}$$

풀이 (1) 대각항 $= 1, -5, 9$ $\quad \mathrm{tr}(A) = 1-5+9 = 5$

(2) 대각항 $= 1, 4$ $\quad \mathrm{tr}(B) = 1+4=5$

(3) 대각항과 대각합은 정방행렬에서만 정의되는데 C는 정방행렬이 아니다. ■

2.4 컴퓨터 프로그램에 의한 연산

Trace(A)

```
>> A=[1, 3, 6; 2, -5, 8; 4, -2, 9];
>> trace(A)

ans =

     5

>>
```

Trace(B)

```
>> B=[1, 2; 3, 4];
>> trace(B)

ans =

     5

>>
```

실습 ❷-12

예제 ②-33 MATLAB

다음에 주어진 행렬 A의 계수를 구해 보자.

$$A = \begin{bmatrix} 1 & 2 & -3 & 1 & 2 \\ 2 & 4 & -4 & 6 & 10 \\ 3 & 6 & -6 & 9 & 13 \end{bmatrix}$$

풀이 그 결과 다음과 같은 행 사다리꼴이 만들어진다.

$$\begin{bmatrix} 1 & 2 & -3 & 1 & 2 \\ 0 & 0 & 2 & 4 & 6 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix}$$

따라서 주어진 행렬의 계수는 3이 된다. ■

2.4 컴퓨터 프로그램에 의한 연산

```
>> A=[1, 2, -3, 1, 2; 2, 4, -4, 6, 10; 3, 6, -6, 9, 13];
>> rank(A)

ans =

     3

>>
```

실습 ❷-13

예제 ②-34

MATLAB

다음 행렬 A의 계수를 구해 보자.

$$A=\begin{bmatrix} 3 & -3 & 0 \\ 1 & 4 & 6 \\ 4 & 6 & -3 \end{bmatrix}$$

풀이

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 5 & 6 \\ 0 & 0 & -15 \end{bmatrix}$$

그 결과 행의 각 항들이 모두가 0이 아닌 행의 개수가 3개이므로 행렬 A의 계수는 3이다. ■

컴퓨터 프로그램에 의한 연산

```
>> A=[3, -3, 0; 1, 4, 6; 4, 6, -3];
>> rank(A)

ans =

     3

>>
```

행렬의 생활속의 응용

- 행렬은 선형방정식을 간단하게 표현할 수 있으며, 더욱 쉽게 연산을 할 수 있도록 해 준다. 예를 들어, 2x + 3y = -5와 -3x + y = 2인 방정식을

$\begin{bmatrix} 2 & 3 & -5 \\ -3 & 1 & 2 \end{bmatrix}$ 와 같이 행렬로 간단히 나타낼 수 있다.

- 경영에 있어서의 재무 분석과 손익 분기점이나 위기 관리에 매우 중요한 역할을 한다.
- 인공위성을 통한 자동위치측정시스템, 즉 GPS(Global Positioning System)에의 응용을 통하여 선박, 항공기, 자동차 등의 행로나 길 안내에 응용된다.
- 컴퓨터 통신에서의 네트워크 관리 문제도 선형방정식을 통한 행렬을 통해 분석할 수 있다.
- 가계의 총 수입과 항목별로 가격과 단위의 다양한 지출 관계를 행렬로 표현하면 효과적인 계산과 현명한 소비를 도와준다.
- 어떤 사회 현상들에 대한 판단을 위한 근사값을 짐작할 수 있고 사회적 통찰의 직관을 넓힐 수 있게 도와준다.