



11 강 다차원 배열 구조

C 언어

학습목표

- ▶ 다차원 배열의 개념에 대해 알 수 있다.
- ▶ 2차원 배열을 이용한 자료를 처리할 수 있다.



EBS tip

1 다차원 배열의 정의

2개 이상의 첨자를 사용하여 나타낸 배열을 말한다.

2차원 : 자료형 배열명[첨자][첨자]
3차원 : 자료형 배열명[첨자][첨자][첨자]
:
n차원 : 자료형 배열명[첨자1]...[첨자n]

2 2차원 배열

- ① 2차원 배열은 행과 열로 구성된 배열을 말한다.
- ② 2차원 배열의 각 요소는 행과 열의 첨자로 표시한다.
- ③ 2차원 배열의 선언

예 2차원 배열 선언의 예

```
int num[2][3];
```

- 배열명은 num이다.
 - 배열의 자료형은 정수형이다.
 - 배열 요소의 개수는 $2 \times 3 = 6$ (개)이다.
- ④ 2차원 배열의 행 첨자와 열 첨자는 모두 0부터 시작하며, 행 방향이 먼저 기억된다.
 - ⑤ 배열의 초기화 방법
 - 각 배열 요소에 대한 초기화(int num[2][3];)
num[0][0]=3, num[0][1]=4, num[0][2]=5;
num[1][0]=6, num[1][1]=7, num[1][2]=8;
 - 행 단위로 초기화
int num[2][3]={ {3, 4, 5}, {6, 7, 8} };
또는 int num[2][3] = {3, 4, 5, 6, 7, 8};
 - ⑥ 2차원 배열의 논리적 구조
int num[2][3];

	0	1	2
0	3	4	5
1	6	7	8

■ 2차원 배열의 실제 메모리 구조

3	num[0][0]
4	num[0][1]
5	num[0][2]
6	num[1][0]
7	num[1][1]
8	num[1][2]

예 배열의 각 요소의 합을 구하는 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main(void){ int a, b, tot; int num[4][4]={ {1, 2, 3, 4}, {2, 3, 4, 5}, {2, 4, 6, 8}, {1, 3, 5, 7}}; tot=0; for(a=0; a<=3; a++) for(b=0; b<=3; b++) tot=tot+num[a][b]; printf("%d", tot); }</pre>	60

[프로그램 해설]

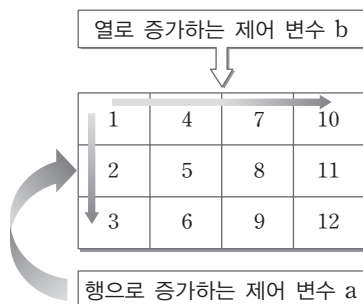
- 배열 num은 4행 4열로 구성되어 있다.
- 다중 for 문에 의해 a = 0일 때, b = 0, 1, 2, 3으로 변환된다. 따라서 배열 num에 있는 0행의 모든 원소가 tot에 더해진다.
- a = 1, a = 2, a = 3일때 반복문을 계속 수행하여 배열 num에 있는 모든 요소 값의 합을 구한다.

3 2차원 배열의 활용

예 2차원 배열 but[3][4]에 규칙적인 숫자 넣는 프로그램

1	4	7	10
2	5	8	11
3	6	9	12

- 숫자를 넣는 방법은 그림과 같다. 즉 행을 먼저 증가하면서 숫자를 1씩 증가시켜 해당 위치에 값을 넣어 준다.



배열 num의 논리적 구조

	0	1	2	3
0	1	2	3	4
1	2	3	4	5
2	2	4	6	8
3	1	3	5	7



프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main(void){ int a, b, n; int buf[3][4]; n=1; for(b=0; b<=3; b++){ for(a=0; a<=2; a++){ buf[a][b]=n; n++; } for(a=0; a<=2; a++){ for(b=0; b<=3; b++){ printf("%d ", buf[a][b]); printf("\n"); } } } }</pre>	<pre>1 4 7 10 2 5 8 11 3 6 9 12</pre>

- 2차원 배열의 경우 다중 반복문을 이용하여 값을 처리하게 된다. 특히 행과 열의 변수값의 변화에 주의하여야 한다.

[프로그램 해설]

배열에 숫자를 넣는 순서를 보면 다음과 같다.

buf[0][0] → buf[1][0] → buf[2][0]

buf[0][1] → buf[1][1] → buf[2][1]

buf[0][2] → buf[1][2] → buf[2][2]

buf[0][3] → buf[1][3] → buf[2][3]

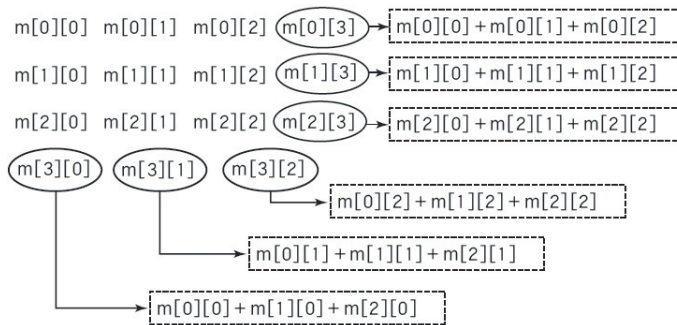
따라서 반복 처리되는 제어 변수를 가지고 알아보면 buf[a][b]는 다음과 같이 나타낼 수 있다.

b=0	a=0	1			
	a=1	2			
	a=2	3			
b=1	a=0	1	4		
	a=1	2	5		
	a=2	3	6		
b=2	a=0	1	4	7	
	a=1	2	5	8	
	a=2	3	6	9	
b=3	a=0	1	4	7	10
	a=1	2	5	8	11
	a=2	3	6	9	12

예 2차원 배열에 저장된 자료의 가로, 세로 합계를 구하는 프로그램

10	20	30	60
40	50	60	150
70	80	90	240
120	150	180	

• 가로, 세로 합을 구하는 과정



프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main(void){ int a, b; int m[4][4]={ {10, 20, 30, 0}, {40, 50, 60, 0}, {70, 80, 90, 0}, {0, 0, 0, 0}; for(a=0; a<=2; a++){ for(b=0; b<=2; b++){ m[a][3]=m[a][3]+m[a][b]; m[3][b]=m[3][b]+m[a][b]; } } for(a=0; a<=3; a++){ for(b=0; b<=3; b++){ printf("%5d", m[a][b]); printf("\n"); } } }</pre>	<pre>10 20 30 60 40 50 60 150 70 80 90 240 120 150 180 0</pre>

[프로그램 해설]

배열 `m[a][b]`에서 가로 합계는 `m[a][3]`에 각각의 행에 있는 요소의 값을 누적하고, 세로 합계도 `m[3][b]`에 각각의 열에 있는 요소값을 누적하면 된다.



■ 배열의 위치 값을 변수가 아닌 상수값으로 지정한 경우, 해당 위치는 상수값에 해당하는 위치로 고정된다.



예 행렬의 덧셈을 구하는 프로그램

행렬의 합을 구하기 위해서는 연산을 수행할 두 행렬의 크기가 같아야 한다.
행렬의 덧셈은 같은 자리의 원소를 서로 더하면 된다.

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main(){ int mat1[2][3] = {{1, 2, 3}, {4, 5, 6}}; int mat2[2][3] = {{7, 8, 9}, {1, 2, 3}}; static mat3[2][3] = {0}; int k, m; for(k = 0; k <= 1; k++){ for(m = 0; m <= 2; m++){ mat3[k][m] = mat1[k][m] + mat2[k][m]; } } for(k = 0; k <= 1; k++){ for(m = 0; m <= 2; m++){ printf("%3d", mat3[k][m]); printf("\n"); } } }</pre>	<pre>8 10 12 5 7 9</pre>

- 행 단위로 값을 초기화하는데 있어서 초기화 리스트의 수가 행 단위 요소의 수보다 적은 경우, 해당 부분은 0으로 초기화된다.

예]

```
int m[3][3]={
    {1},
    {2, 3},
    {4, 5, 6}};
```

	0	1	2
0	1	0	0
1	2	3	0
2	4	5	6

[프로그램 해설]

- mat1[2][3]은 2행 3열의 행으로 이루어져 있으며, 논리적인 형태는 다음과 같다.

	0	1	2
0	1	2	3
1	4	5	6

- mat2[2][3]은 2행 3열의 행으로 이루어져 있으며, 논리적인 형태는 다음과 같다.

	0	1	2
0	7	8	9
1	1	2	3

- 반복문을 이용하여 mat1 배열의 요소와 mat2 배열 요소의 합을 구해 mat3 배열에 저장하는 형태는 다음과 같다.

mat1[2][3]		mat2[2][3]												
<table> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> </table>	1	2	3	4	5	6	+	<table> <tr><td>7</td><td>8</td><td>9</td></tr> <tr><td>1</td><td>2</td><td>3</td></tr> </table>	7	8	9	1	2	3
1	2	3												
4	5	6												
7	8	9												
1	2	3												
↓														
mat3[2][3]		<table> <tr><td>8</td><td>10</td><td>12</td></tr> <tr><td>5</td><td>7</td><td>9</td></tr> </table>	8	10	12	5	7	9						
8	10	12												
5	7	9												



기출 모의고사

정답 및 해설 p. 12

1

2007학년도 대수능

다음 프로그램의 실행 결과를 구한 후, [프린터 출력 예]와 같이 변환했을 때 프린터 출력 형태로 옳은 것은?

```
#include <stdio.h>
void main(){
    int x, y;
    int pflag[3][2]={0,0,
                     {1,1},
                     {0,1}};

    y = 0;
    while(y < 2){
        for(x=2; x>=0; x--){
            printf("%d", pflag[x][y]);
        }
        printf("\n");
        y++;
    }
}
```

[프린터 출력 예]

프로그램 실행 결과인 (가)를 프린터에 전송하면 (나)와 같이 출력된다.

(가) 프로그램 실행 결과

0	0
1	1
0	1



(나) 프린터 출력 형태

- ①

0	0
1	1
0	1

 ②

0	0
1	1
0	1

 ③

0	0
1	1
0	1
- ④

0	0
1	1
0	1

 ⑤

0	0
1	1
0	1

2

2010년 6월 시행 평가원 모의평가

다음 프로그램은 그레이 영상을 흑백 영상으로 바꾸는 과정이다. 프로그램의 실행 결과를 [예시]와 같이 표현할 때 모니터 화면 형태로 옳은 것은?

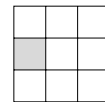
```
#include <stdio.h>
void main() {
    int a, b;
    int img[3][3] = {{129,125,255},
                     {0,255,63},
                     {130,127,128}};

    int result[3][3];
    for(a=0; a<3; a++) {
        for(b=0; b<3; b++) {
            if (img[a][b] > 127)
                result[a][b] = 1;
            else
                result[a][b] = 0;
            printf("%d", result[a][b]);
        }
        printf("\n");
    }
}
```

[예시]

0	0	0
1	0	0
0	0	0

실행 결과



모니터 화면 형태

- ①

0	0	0
1	0	0
0	0	0

 ②

0	0	0
1	0	0
0	0	0

 ③

0	0	0
1	0	0
0	0	0
- ④

0	0	0
1	0	0
0	0	0

 ⑤

0	0	0
1	0	0
0	0	0

3 다음 프로그램의 출력 결과는?

```
#include <stdio.h>
void main(){
    int a, b;
    int sum = 0;
    int num[3][3]={1, 2, 3, 4, 5, 6, 7, 8, 9};
    for(a=0; a<=2; a++){
        for(b=0; b<=2; b++){
            if (a%2 == 0)
                sum = sum + num[a][b];
        }
    }
    printf("%2d", sum);
}
```

- ① 20 ② 25 ③ 30
④ 35 ⑤ 40

4 다음은 오류 비트를 검사하기 위한 프로그램의 일부이다. 실행 결과는?

```
#include <stdio.h>
void main(){
    int k, m, v, h, c1, c2;
    int digit[4][4] = {{1,0,0,1},
                       {1,0,1,0},
                       {1,1,1,0},
                       {1,0,0,1}};

    c1 = 0, c2 = 0;
    for(k = 0; k <= 3; k++){
        v = 0, h = 0;
        for(m = 0; m <= 3; m++){
            v = v + digit[k][m];
            h = h + digit[m][k];
        }
        if (v % 2 != 0)
            c1 = k;
        if (h % 2 != 0)
            c2 = c2 + 1;
    }
    printf("%2d %2d", c1, c2);
}
```

- ① 1 1 ② 1 2 ③ 2 1
④ 3 1 ⑤ 3 2

5 다음 프로그램을 실행했을 때 배열 data의 값으로 옳은 것은? (단, a는 행, b는 열을 나타내는 변수)

```
#include <stdio.h>
void main() {
    int data[3][3];
    int a, b;

    for(a=0; a<=2; a++) {
        for(b=0; b<=2; b++) {
            if (a==b)
                data[a][b]= a + 1;
            else
                data[a][b]= b + 1;
        }
    }

    for(a=0; a<=2; a++) {
        for(b=0; b<=2; b++)
            printf("%d ", data[a][b]);
        printf("\n");
    }
}
```

- ①

	0	1	2
0	1	2	3
1	1	2	3
2	1	2	3

 ②

	0	1	2
0	1	1	1
1	2	2	2
2	3	3	3
- ③

	0	1	2
0	0	0	1
1	0	1	0
2	1	0	0

 ④

	0	1	2
0	0	0	1
1	0	2	0
2	3	0	0
- ⑤

	0	1	2
0	1	0	1
1	0	2	0
2	3	0	3