C 프로그래밍 및 실습

[13주차 1강]
배열 (4)

학습 내용

7.4 다차원 배열

8.5 함수에 배열 전달(다차원 배열)

학습 목표

7.4 배열의 선언, 사용, 초기화 방법을 익힌다.

8.5 함수에 다차원 배열을 전달하는 방법을 익힌다.

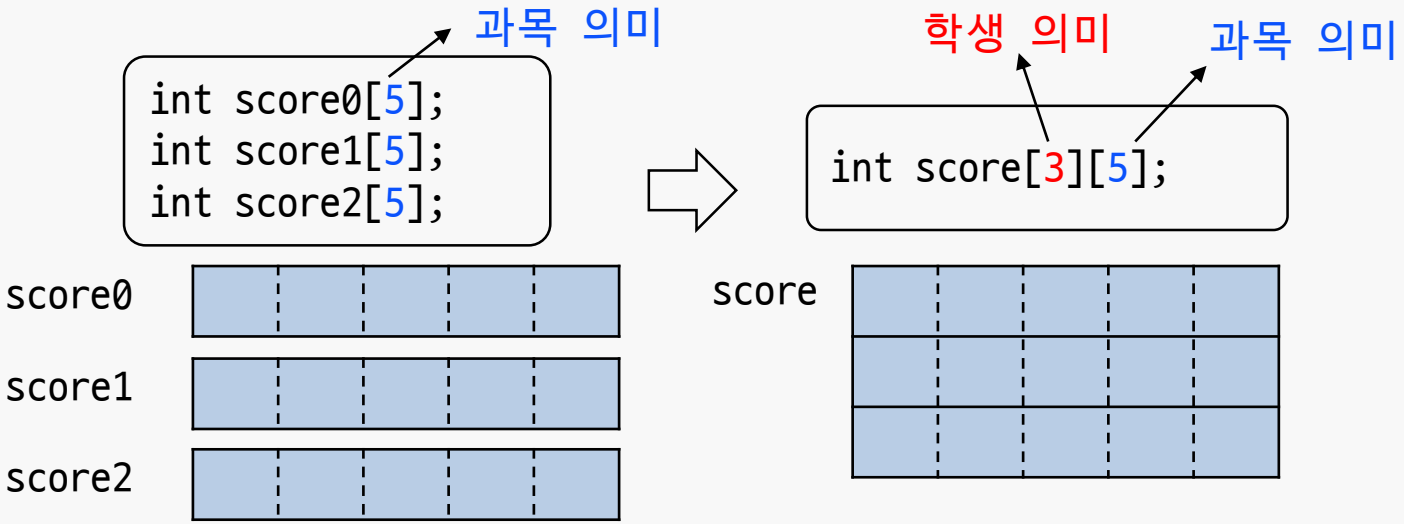
C 프로그래밍 및 실습

7.4 다차원 배열

세종대학교

'3명 학생'의 '5개 과목 성적'을 처리해야 한다면?

- '**정수**'를 여러 개 묶어 '**정수 배열**'을 만들었듯이
- '**정수 배열**'을 여러 개 묶어 '**정수 배열의 배열**'을 만들면 효과적



- **다차원 배열** : 첨자가 두 개 이상인 배열 (2차원 배열, 3차원 배열 등)

세종대학교

2차원 배열의 선언

- 구문

```
자료형 변수명[크기][크기];
```

- 예시)

✓ `int x[3][5];` ⟶

- 자료형은 정수형, 이름은 x, 크기는 3x5 인 2차원 배열
- 총 3x5=15개의 정수 저장

✓ `int x[3][5], y[7][6];` ⇨ 다양한 크기의 배열 함께 선언

✓ `double a, b, c[10], d[9][9];` ⇨ 다른 차원 배열도 함께 선언
(코드 가독성 나빠질 수 있음)

세종대학교

2차원 배열의 원소

- 두 개의 첨자를 이용하여 원소 표현
- 각 차원의 첨자는 0부터 시작
- 예시) `int x[3][5];  // 선언`
  - ✓첫번째 차원의 첨자는 0~2
  - ✓두번째 차원의 첨자는 0~4

	0	1	2	3	4
0	x[0][0]	x[0][1]	x[0][2]	x[0][3]	x[0][4]
1	x[1][0]	x[1][1]	x[1][2]	x[1][3]	x[1][4]
2	x[2][0]	x[2][1]	x[2][2]	x[2][3]	x[2][4]

- 2차원 배열의 각 원소는 하나의 일반 변수와 동일하게 취급

```
x[1][2] = 10;
x[0][3] = x[0][3] + 5;
printf("x[1][4] = %d", x[1][4] );
```

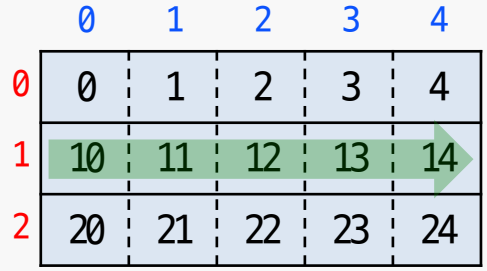
2차원 배열과 반복문

- 선언) int score[3][5]; //3명의 학생, 5개의 과목 점수 저장
- 1번 학생의 모든 과목 점수 출력

```
for( j=0; j < 5 ; ++j )
   printf(" %d", score[1][j]);
```

실행 결과

```
10 11 12 13 14
```

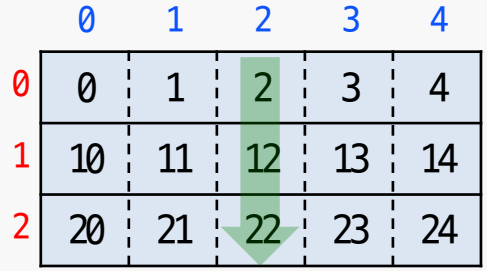


- 2번 과목의 모든 학생 점수 출력

```
for( i=0; i < 3 ; ++i )
   printf(" %d", score[i][2]);
```

실행 결과

```
2 12 22
```

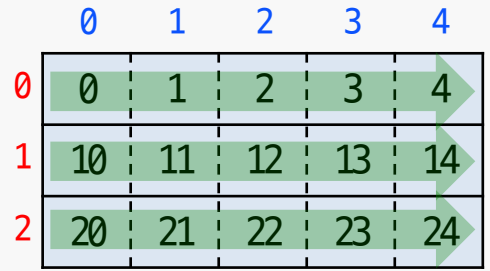


세종대학교

2차원 배열과 반복문

- 각 학생별로 모든 과목 점수 출력
  - ✓즉, 학생을 기준으로 모든 점수 처리

```
for( i=0; i < 3 ; ++i ) {
    for( j=0; j < 5 ; ++j )
        printf(" %d", score[i][j]);
    printf("\n");
}
```



실행 결과

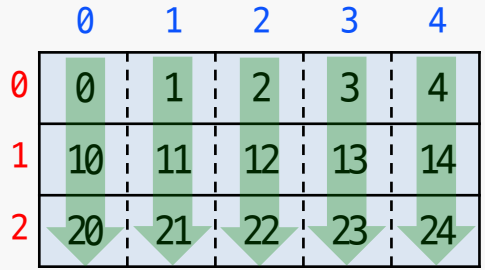
```
0 1 2 3 4
10 11 12 13 14
20 21 22 23 24
```

세종대학교

2차원 배열과 반복문

- 각 과목별로 모든 학생 점수 출력
  - ✓즉, 과목을 기준으로 모든 점수 처리

```
for( j=0; j < 5 ; ++j ) {
    for( i=0; i < 3 ; ++i )
        printf(" %d", score[i][j]);
    printf("\n");
}
```



실행 결과

```
0 10 20
1 11 21
2 12 22
3 13 23
4 14 24
```

2차원 배열 초기화

	0	1	2	3	4
0	10	20	30	0	0
1	40	50	60	70	0
2	0	0	0	0	0

- 중첩 중괄호를 사용하여 행별로 초깃값 설정
- 값이 지정되지 않은 원소는 0으로 초기화
  - ✓한 줄에 쓴 형태

```
int x[3][5] = { {10, 20, 30}, {40, 50, 60, 70} } ;
```

  - ✓행 별로 나눠 쓴 형태

```
int x[3][5] = { {10, 20, 30}, // 0번 행
                {40, 50, 60, 70} } ; // 1번 행
```

- 참고) 하나의 중괄호를 사용하는 형태도 가능(교재 p.193 참조)

세종대학교

배열 전체를 0으로 초기화 하기

- `int x[3][5] = { { 0 } };` ⇨ 중첩 중괄호 사용 형태
- `int x[3][5] = { 0 };` ⇨ 단일 중괄호 사용 형태

배열의 크기를 초깃값 개수로 정하기

- 첫번째 첨자만 생략 가능, 두번째 첨자는 생략 불가능
- `int x[ ][2] = { {0,1}, {0}, {0} };` ⇨ x[3] [2]와 동일 (정상)
- `int x[3][ ] = { {0,1}, {0}, {0} };` ⇨ (X) 컴파일 오류
- `int x[ ][ ] = { {0,1}, {0}, {0} };` ⇨ (X) 컴파일 오류

실습 하기

[예제 7.6] 학생 3명의 국어, 영어 성적을 저장하기 위한 2차원 배열을 선언하고 아래와 같이 초기화 한 후, 국어와 영어 과목의 평균을 각각 계산하여 출력하기

	국어	영어
학생 A	20	100
학생 B	70	36
학생 C	30	50

실행 결과

```
40.000000          ⇨ 국어 평균
62.000000          ⇨ 영어 평균
```

실습 하기

[예제 7.7]
4×4 크기의 행렬을 나타내는 2차원 배열 A를 아래 왼쪽과 같이 초기화 하고, 행렬 A와 A의 전치 행렬을 나란히 출력

- ✓ 전치 행렬이란? 행과 열을 바꾼 행렬
- ✓ 전치 행렬을 저장하기 위한 배열을 따로 선언해도 무방

실행 결과

```
0.0  0.1  0.2  0.3        0.0  1.0  2.0  3.0
1.0  1.1  1.2  1.3        0.1  1.1  2.1  3.1
2.0  2.1  2.2  2.3        0.2  1.2  2.2  3.2
3.0  3.1  3.2  3.3        0.3  1.3  2.3  3.3
```

(행렬 A) (A의 전치 행렬)

- (응용) 전치 행렬을 저장하기 위한 배열을 따로 선언하지 않고 A를 이용하여 직접 전치 행렬 출력하기

세종대학교

3차원 이상의 배열

- 2차원 배열과 비슷한 방법으로 확장
- 예시) 반별로 학생 3명의 국어와 영어 성적 처리

1반	국어	영어
학생 A	20	90
학생 B	70	36
학생 C	30	50

2반	국어	영어
학생 D	30	90
학생 E	80	40
학생 F	40	60

✓ `int score[2][3][2] =`

```
{ { {20,90}, {70,36}, {30,50} },    // 1반
  { {30,90}, {80,40}, {40,60} } };  // 2반
```

3차원 이상의 배열

- 두 반의 국어 성적 전체 출력하기

```
int i, j;
int score[2][3][2] = { { {20,90},{70,36},{30,50} },
                       { {30,90},{80,40},{40,60} } };

for( i=0 ; i<2 ; i++ ) {   // 각 반에 대해서
   printf("%d반 국어 :", i+1);
   for( j=0 ; j<3 ; j++ )  // 각 학생에 대해서
      printf(" %d", score[i][j][0]);
   printf("\n");
}
```

실행 결과

```
1반 국어 : 20 70 30
2반 국어 : 30 80 40
```

C 프로그래밍 및 실습

8.5 함수에 배열 전달(다차원 배열)

함수에 일차원 배열 전달하기

- 배열 크기 명시해도 의미 없음 → 보통 생략
- 배열 크기가 필요한 경우, 별도의 함수 인자로 전달

```
void print_arr( int x[ ], int n ) {
   int i;
   for( i=0; i < n ; ++i )
      printf(" %d", x[i]);   // 배열의 원소 값 출력
   printf("\n");
}
int main() {
   int a[3] = { 10, 20, 30 };

   print_arr( a , 3 );       // 배열 a와 a의 크기 전달

   return 0;
}
```

세종대학교

함수에 다차원 배열 전달하기

- 첫번째 첨자의 크기는 의미 없음 → 보통 생략
- 두번째 이후 첨자는 명시 필요

```
void print_arr2( int x[  ][5], int n ) {
   int i, j;
   for( i=0; i < n ; ++i ) {   // 각 행에 대해
      for( j=0; j < 5 ; ++j )  // 각 열의 원소 값 출력
         printf(" %d", x[i][j]);
      printf("\n");
   }
}
int main() {
   int a[3][5] = { {10, 20, 30}, {40, 50} };

   print_arr2( a , 3 );        // 배열 a와 첫 번째 첨자의 크기 전달

   return 0;
}
```

열의 크기가 5인
2차원 배열 전달

학습 정리

- 첨자가 하나인 배열을 **일차원 배열**, 두 개 이상인 배열을 **다차원 배열**이라고 함
- **2차원 배열**을 선언하거나 원소 값을 참조를 위해서 두 개의 첨자를 사용함
- **2차원 배열**의 초기화에는 보통 중첩된 중괄호를 사용함
- 함수의 형식인자에서 **다차원 배열**의 첫번째 첨자는 보통 생략함