

---

# C 프로그래밍 및 실습

## 7. 배열

세종대학교

---

# 목차

---

- 1) 배열 개요
- 2) 배열 선언과 사용
- 3) 배열 활용 예제
- 4) 다차원 배열

# 1) 배열 개요

---

- 변수를 여러 개 만들어야 하는 상황을 생각해 보자.
  - 사용자로부터 5개의 정수를 입력 받아 변수에 저장하고, 이 값을 출력하는 프로그램은 다음과 같이 작성할 수 있다.
  - 하지만, 정수가 100개라면? → 배열을 사용하여 해결

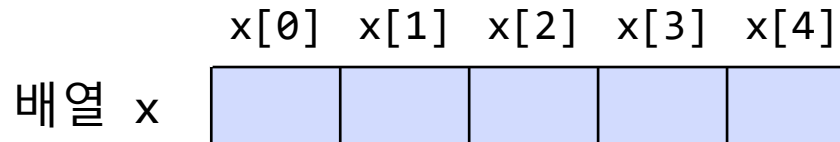
```
int main(void){  
  
    int x0, x1, x2, x3, x4;  
  
    scanf("%d%d%d%d%d", &x0, &x1, &x2, &x3, &x4);  
  
    printf("%d %d %d %d %d\n", x0, x1, x2, x3, x4);  
  
    return 0;  
  
}
```

# 1) 배열 개요

---

- 배열이란?

- 같은 자료형의 변수 여러 개를 하나로 묶은 자료형
- 배열을 이용하여 많은 변수를 한번에 선언하고, 저장된 데이터를 처리할 수 있음
- 앞 예에서 사용한 5개의 변수를 배열 형식으로 표현하면  
✓  $x_0, x_1, x_2, x_3, x_4 \rightarrow x[0], x[1], x[2], x[3], x[4]$



## 2) 배열 선언과 사용

### 배열의 선언

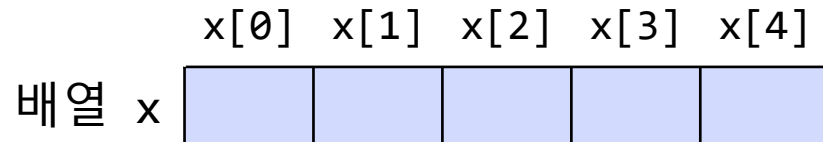
- 변수이름 뒤에 필요한 변수의 개수(배열 크기)를 명시
- 배열 선언 구문

자료형 변수명[ 배열크기 ];

### 예시)

✓ `int x[5];`

- 자료형은 정수형, 이름은 x, 크기는 5 인 배열
- 5개의 정수를 저장하는 배열 x



✓ `int x[5], y[3];`

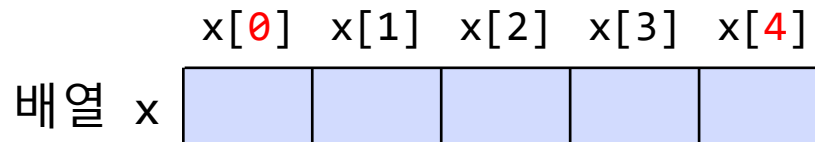
⇒ 여러 개의 배열 함께 선언 가능

✓ `double a, b, c[10];`

⇒ 일반 변수와 함께 선언 가능

## 2) 배열 선언과 사용

- 배열의 **원소** (or 요소): 배열을 구성하는 각 변수를 지칭
  - 대괄호 [ ] 안에 번호를 넣어서 구분: x[0], x[1], x[2], x[3], x[4]
- 배열의 **첨자** or **인덱스**(index)
  - ✓ 배열에서 각 원소의 위치를 나타내는 대괄호 [ ] 안의 번호를 지칭
  - ✓ 배열의 인덱스는 **0부터** 시작함
  - ✓ 예) 크기가 5인 배열의 인덱스는 0~4까지 임



- 배열의 각 원소는 하나의 일반 변수와 동일하게 취급

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

```
a = 10;  
b = b + 3;  
printf("%d", c );
```

## 2) 배열 선언과 사용

- 앞서 본 예제 프로그램을 배열을 이용하여 작성해보자.
  - 사용자로부터 5개의 정수를 입력 받아 변수에 저장하고, 이 값을 출력하는 프로그램

```
int x[5];  
scanf("%d%d%d%d%d", &x[0], &x[1], &x[2], &x[3], &x[4]);  
printf("%d %d %d %d %d\n", x[0], x[1], x[2], x[3], x[4]);
```

실행 예시

3	5	1	-3	4
3	5	1	-3	4

x[4]를 하나의 변수처럼 사용  
scanf에서 변수 앞에 & 붙여줌

✓ 부동소수형, 문자형에 대해서도 동일한 방법으로 배열 선언 및 사용

## 2) 배열 선언과 사용

- 앞 프로그램에서 입력되는 정수가 100개라면?
  - 배열을 사용해서 변수 선언은 간단히 해결됨
  - 하지만, 입출력 부분은? → 반복문을 이용하여 해결
- 배열과 반복문과의 만남
  - 배열 원소의 인덱스가 0부터 시작하여 1씩 증가한다는 규칙을 이용하여 배열의 원소에 접근
    - ✓ 참고) 배열 첨자로 결과 값이 정수인 수식은 모두 가능

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



```
for( i=0 ; i<5 ; i++ )  
    printf("%d ", x[i]);
```



## 2) 배열 선언과 사용

- 배열을 이용하여 작성된 최종 프로그램
  - 사용자로부터 5개의 정수를 입력 받아 변수에 저장하고, 이 값을 출력하는 프로그램

```
int x[5], i;    // 배열 선언

for( i=0 ; i<5 ; i++ )
    scanf("%d", &x[i]); // 입력된 정수를 배열에 저장

for( i=0 ; i<5 ; i++ )
    printf("%d ", x[i]); // 배열에 저장된 정수 출력
printf("\n");
```

### ※ 배열 사용시 주의점

컴파일러는 배열의 첨자가 유효한 범위인지 검사하지 못함  
유효하지 않은 첨자 범위 → 런타임 오류를 유발시킴

## 2) 배열 선언과 사용

### [실습1]

- 크기가 7인 배열 x에 아래 점수를 저장하시오. **for문**

80, 71, 91, 95, 77, 79, 88

- for 문을 이용하여 80점 이상의 학생의 인덱스와 점수를 모두 출력하는 프로그램을 작성하시오.

출력 예시

0	80
2	91
3	95
6	88

### [실습2]

- 크기가 9인 배열 x를 선언하시오.
- 구구단 3단의 계산 값을 배열에 저장한 후, **for문**
- 배열 내용을 화면에 출력하시오. **for문**

출력 예시

3
6
9
.
.
.
27

## 2) 배열 선언과 사용

### 배열 초기화

- 배열 전체를 선언과 동시에 초기화 하기
- 중괄호 `{ }` 안에 배열이 초기화 값을 **숫자**로 구분하여 나열
- 배열 전체 값을 한꺼번에 대입하는 것은 선언 시에만 가능

```
int i, x[5] = {0, 10, 20, 30, 40} ;  
  
for(i=0;i<5;i++)  
    printf(" %d", x[i]);
```

실행 결과

```
0 10 20 30 40
```

## 2) 배열 선언과 사용

- 배열의 크기보다 초기값의 개수가 작으면?

- 앞 원소부터 차례로 채워지고, **배열의 뒷부분은 0으로** 채워짐

```
int i, x[5] = {1, 2, 3} ;  
for(i=0;i<5;i++)  
    printf(" %d", x[i]);
```

실행 결과

1 2 3 0 0

- 배열 전체를 0으로 초기화 하기

```
int i, x[5] = {0} ;  
for(i=0;i<5;i++)  
    printf(" %d", x[i]);
```

```
int i, x[5] = {0,} ;  
for(i=0;i<5;i++)  
    printf(" %d", x[i]);
```

- 주의) 배열의 크기보다 초기값의 개수가 크면?

✓ int x[5] = { 1, 2, 3, 4, 5, 6 };      ⇨ **컴파일 오류** 발생

## 2) 배열 선언과 사용

- 배열의 크기를 초기값의 개수로 정하기

- 배열의 크기를 지정하지 않으면?

→ 배열 크기가 초기화에 사용된 원소 수로 결정됨

```
int x[ ] = {10, 20, 30};
```

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

```
printf("\n배열 크기 = %d\n", sizeof(x)/sizeof(x[0]));
```

실행 결과

10 20 30  
배열 크기 = 3

- ✓ sizeof 연산자 (2장 참조)

- ✓ sizeof(x) : 배열 x의 전체 크기

- ✓ sizeof(x[0]) : 원소 하나의 크기

# 목차

---

- 1) 배열 개요
- 2) 배열 선언과 사용
- 3) **배열 활용 예제**
- 4) 다차원 배열

### 3) 배열 활용 예제

- [예제 7.1] 5개의 실수를 입력 받아 배열에 저장하고, 합과 평균 출력  
✓ 참고) scanf() 함수에서 double 형 서식 지정자 : %lf

입력 예시1

1.0 2.0 3.0 4.0 5.0

출력 예시1

합 = 15.000000, 평균 = 3.000000

- [예제 7.2] 5개의 실수를 입력 받아 배열에 저장하고, 최댓값 출력

입력 예시1

5.0 0.7 13.4 4 9.5

출력 예시1

max = 13.400000

- (응용) 최댓값이 저장된 배열 원소의 첨자 출력하기  
✓ 위 예에서는 2

### 3) 배열 활용 예제

- **[예제 7.3]** 크기가 10인 두 배열 X와 Y에 정수를 입력 받아서 저장한 후, X[0]과 Y[9]의 합을 Z[0]에 저장하고, X[1]과 Y[8]의 합을 Z[1]에 저장하고, ..., X[9]와 Y[0]의 합을 Z[9]에 저장한 후, 배열 Z의 원소를 출력하자.
  - ✓  $z[0] = x[0] + y[9] = 1 + 9 = 10$
  - ✓  $z[1] = x[1] + y[8] = 2 + 7 = 9$
  - ✓ hint) 배열 x, y, z의 첨자가 변화는 규칙을 찾아보라.
  - ✓ 참고) 첨자에는 결과 값이 정수인 어떤 수식이든 사용 가능

입력 예시1

1	2	3	4	5	6	7	8	9	10
2	4	6	8	0	1	3	5	7	9

출력 예시1

10	9	8	7	7	6	6	15	14	13	12
----	---	---	---	---	---	---	----	----	----	----



### 3) 배열 활용 예제

---

- **[예제 7.4]** 9개의 영어 소문자를 입력 받아 배열에 저장하고, 입력 문자 순서대로 각 문자가 나타난 횟수 출력
  - 예) levelfive → l 2번, e 3번, v 2번, e 3번, ...
  - hint) 이중 반복

입력 예시1

levelfive

출력 예시1

2 3 2 3 2 1 1 2 3

### 3) 배열 활용 예제

- **[예제 7.5]** 0~5 사이의 정수를 10개 입력 받아 배열 x(크기 10)에 저장하고, 0~5 사이의 수가 각각 몇 개인지 계산하여 배열 cnt(크기 6)에 저장한 후, cnt에 저장된 값 출력하기
  - 예시 설명) 0은 1번, 1은 1번, 2는 4번, 3은 0번, 4는 1번, 5는 3번

입력 예시1

2 4 2 5 2 1 5 5 0 2

출력 예시1

1 1 4 0 1 3

- hint) 프로그램 동작 방식
  - ✓ 버전 A: 각 정수마다 배열 x 전체를 훑어서 계산 (배열 x를 총 6번 훑어야 함)
  - ✓ 버전 B: 배열 x를 훑으면서, 각 배열 원소에 해당하는 정수의 cnt 증가 (배열 한 번만 훑어도 됨: 더 효율적)

# 목차

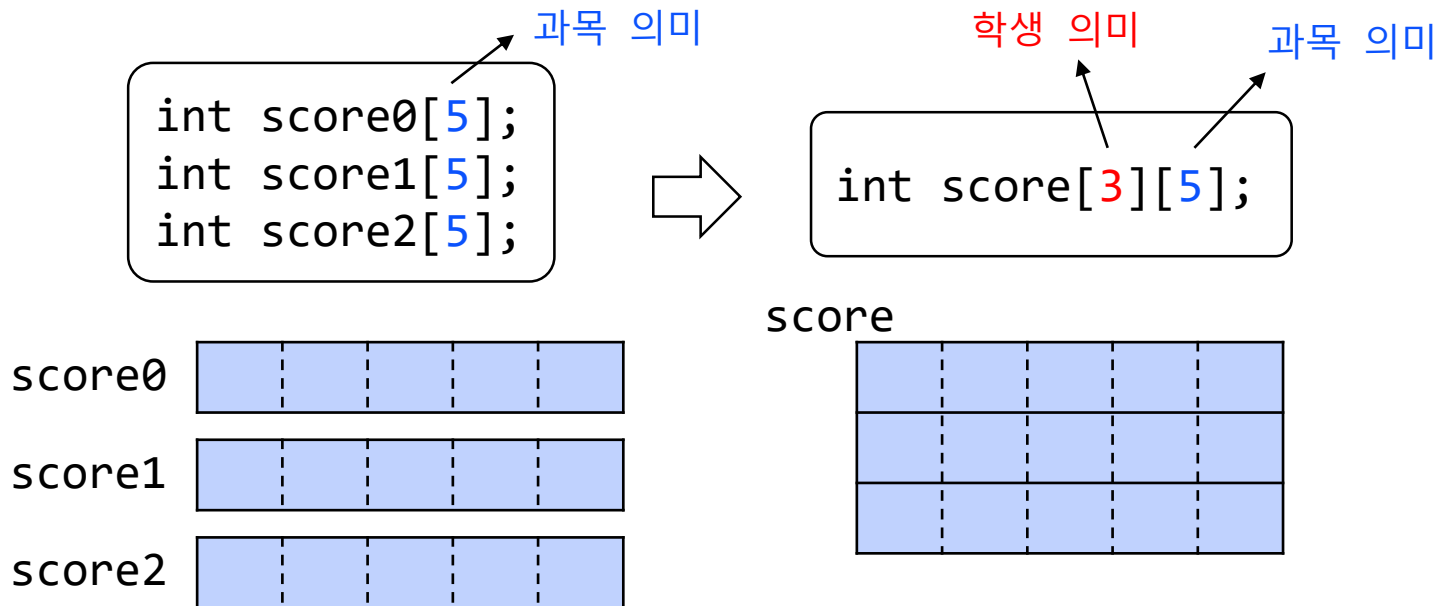
---

- 1) 배열 개요
- 2) 배열 선언과 사용
- 3) 배열 활용 예제
- 4) **다차원 배열**

## 5) 다차원 배열

- '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)  
cf. 일차원 배열

## 5) 다차원 배열

- 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)

## 5) 다차원 배열

### ▪ 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] );
```

## 5) 다차원 배열

### ▪ 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

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

- 2번 과목의 모든 학생 점수 출력

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

실행 결과

2 12 22

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

## 5) 다차원 배열

- 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
```

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



## 5) 다차원 배열

- 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
```

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

## 5) 다차원 배열

### ▪ 2차원 배열 초기화

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

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

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

- ✓ 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)

## 5) 다차원 배열

---

- 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

## 5) 다차원 배열

- **[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

## 5) 다차원 배열

- **[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
1.0	1.1	1.2	1.3
2.0	2.1	2.2	2.3
3.0	3.1	3.2	3.3

(행렬 A)

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

(A의 전치 행렬)

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

## 5) 다차원 배열

### ▪ 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`

## 5) 다차원 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
```