

학습목표

- ▶ 배열의 개념에 대해 알 수 있다.
- ▶ 1차원 배열의 선언 및 사용법에 대해 알 수 있다.

1 배열의 개념

- ① 여러 개의 기억 장소들을 하나의 묶음으로 표현한 자료 구조를 배열이라 한다.
- ② 같은 형의 데이터를 기억시킬 변수에 일일이 변수명을 부여하는 대신에 이를 하나의 집단으로 보고 이 집단에 이름을 정하고, 첨자를 붙여 각각의 변수를 구별한다.
- ③ 배열은 사용하기 전에 자료형, 배열명과 차원, 배열의 크기 등을 미리 선언한다.
- ④ 배열 선언의 일반 형식

자료형 배열명[요소의 수][요소의 수];

- 자료형은 int, float, double, char 등과 같이 C 언어에서 사용되는 모든 자료형을 사용할 수 있다.
- 대괄호 []는 배열을 의미한다.
- []의 개수는 차원을 의미한다.
- [] 안의 숫자(요소의 수)는 주어진 자료형의 크기로 할당되는 기억 장소의 수를 나타내며, 이 수를 첨자라 한다.

2 1차원 배열

- ① 첨자가 하나만 있는 배열을 의미한다.
- ② 배열 선언시 [] 속에 기술하는 요소의 수는 배열 요소의 개수이다.
- ③ 배열 요소는 0부터 시작하여 [배열 크기-1]까지 사용할 수 있다.
- ④ 1차원 배열의 선언

int age[4];

- 배열명은 age이다.
- 배열의 자료형은 int이다.
- 기억 공간의 개수는 4개이다.
- 배열 a는 1차원 배열로 배열 요소는 다음과 같이 4개이다.

age[0]	age[1]	age[2]	age[3]
--------	--------	--------	--------

※ 배열 요소의 첨자는 0부터 시작한다.



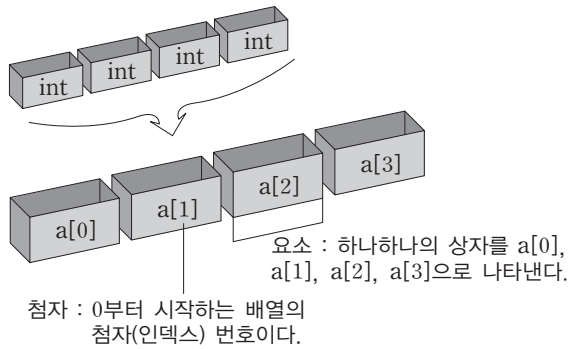
EBS tip

● 배열

- ① 동일한 자료형을 갖는 연속된 자료의 모임
- ② 인덱스(첨자)와 값의 쌍으로 이루어짐.

● 배열의 차수

- ① int num[5]; 1차원 배열
- ② int num[5][5]; 2차원 배열



C 언어의 경우 배열의 첫번째 인덱스 값은 0이다.

⑤ 배열 선언과 동시에 초기화하는 방법

예 int m[4]={5, 2, 3, 4};

m[0]	m[1]	m[2]	m[3]
5	2	3	4

예 int m[4]={1,2};

m[0]	m[1]	m[2]	m[3]
1	2	0	0

⑥ 1차원 배열에서 배열의 크기([] 안의 요소의 개수)를 생략하면 중괄호 안에 자료가 몇 개 있는지에 따라 자동으로 크기가 결정된다.

예 int m[] = { 1, 3, 5, 7, 9};

m[0]	m[1]	m[2]	m[3]	m[4]
1	3	5	7	9

3 배열과 반복문

① int m[5]; 배열에 자료를 입력하는 방법

```
for(a = 0; a <= 4; a++)
    scanf("%d", &m[a]);
```

- 제어 변수 a를 배열의 첨자로 사용한다.
- a = 0 scanf("%d", &m[0]);
- a = 1 scanf("%d", &m[1]);
- a = 2 scanf("%d", &m[2]);
- a = 3 scanf("%d", &m[3]);
- a = 4 scanf("%d", &m[4]);
- 제어 변수 a의 변화에 따라 배열 m에 값을 저장할 수 있다.

② 배열 요소의 접근

```
int main(){
    int k[5];
    k[0] = 20;
    k[1] = 30;
    k[2] = 50;
    k[3] = 70;
    k[4] = 90;
    :
}
```

- 5개의 기억 공간을 갖는 정수형 자료형을 갖는 배열 k를 선언한다.
- k[0]=20;의 의미는 “k라는 배열의 첨자(인덱스) 0에 해당하는 요소에 값 20을 대입하라”는 의미이다.

```
k[0] = 20;    // 첨자값이 0인 요소에 20을 대입한다.
```

- 배열 k에 대입된 자료의 형태

첨자	0	1	2	3	4
요소값	20	30	50	70	90

③ 반복문을 이용하여 배열에 기억된 자료에 접근하는 방법

```
int k[5] = {2, 3, 5, 7, 9};
for(a=0; a <= 4; a++)
    printf("%2d", k[a]);
```

- 실행 과정

a	k[a]	출력값
0	2	2
1	3	3
2	5	5
3	7	7
4	9	9

- for 문을 이용하여 배열에 있는 자료의 개수만큼 반복문을 수행한다.
- 변수 a를 배열 k의 인덱스 값(첨자)으로 사용하여 배열의 요소값을 가져온다.



EBS tip

- 배열의 길이를 선언할 때는 반드시 상수를 사용해야 한다. 즉 다음과 같은 경우 컴파일 오류가 발생한다.

```
void main(){
    int size = 5;
    int m[size];
    :
}
```

왜냐하면 size가 변수이기 때문에 오류가 발생한다.

예 배열에 저장되어 있는 원소의 합을 구하는 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main(){ int m[5] = {1, 2, 3, 4, 5}; int k, sum = 0; for(k = 0; k <= 4; k++){ sum = sum + m[k]; printf("배열 원소의 합 : %d", sum); } }</pre>	배열 원소의 합 : 15

[프로그램 해설]

- 배열 m에 저장된 값은 1, 2, 3, 4, 5이다.
- 반복문을 이용하여 배열 원소의 값을 변수 sum에 더한다. 즉 제어 변수 k가 0인 경우 배열 m[0]의 값인 1을 가져와 변수 sum에 더한다.
- sum값을 출력한다.

예 배열에 있는 자료 중 짝수의 합을 구하는 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main(){ int data[8] = {2, 3, 8, 7, 5, 4, 1, 6}; int k, sum = 0; for(k = 0; k <= 7; k++){ if (data[k] % 2 == 0) sum = sum + data[k]; } printf("%2d", sum); }</pre>	20

- 반복문을 이용하여 배열의 요소를 하나씩 가져온다.
- 배열의 값이 2로 나누어 나머지가 0이면 변수 sum에 누적한 후, sum값을 출력한다.

4 문자 배열

- C 언어에서 문자열은 문자의 집합으로 나타낸다.
- 문자 배열을 사용하기 위해서는 배열명 앞에 char의 자료형을 선언하여야 한다.
- 문자열을 저장하는 배열의 크기는 문자열의 길이에 null 문자('\0') 하나를 더한 크기만큼 지정한다.
- 수치 배열의 초기화 방법과 마찬가지로 배열 선언 후, 그 옆에 특정값으로 초기화한다.

예 char a[4]={"kor"};
char a[]="kor";



● 실행 과정

	0	1	2	3	4
m:	1	2	3	4	5

k	m[k]	sum
0	1	1
1	2	1+2
2	3	1+2+3
3	4	1+2+3+4
4	5	1+2+3+4+5

● 문자열 변수 선언의 경우

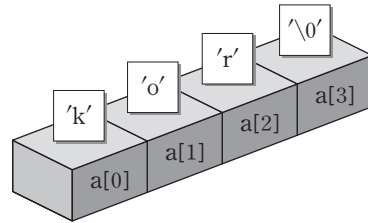
일반적으로 배열의 크기를 명시적으로 선언하지 않아도 된다.

char m[] = "EBSPROGRAM";
배열의 크기가 명시적으로 선언되지 않으면 초기값을 대상으로 자동적으로 계산되기 때문이다.



● C언어에서의 문자열

문자열은 문자열의 끝에 '널(null)' 문자를 갖게 된다.



▲ 1차원 배열에 초기화

- ⑤ 단일 문자 상수로 초기화할 경우는 작은 따옴표(' ')를 사용하고, 여러 개의 단일 문자 상수로 이루어진 문자열인 경우에는 큰 따옴표(" ")를 사용한다. 그러나 배열 명 앞의 자료형 선언은 char로 동일하게 선언한다.

예 char bb[]={'a', 'b', 'c', '\0'};
char bb[]="abc";

- ⑥ 문자 배열의 내용을 출력할 경우 printf() 함수에서 형식 지정 문자는 %c나 %s를 사용한다.

예 문자열에서 같은 문자의 개수가 가장 큰 값을 출력하는 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main(){ char mun[6]="hello"; int a, cnt, b, top=0; for(a=0; a<=4; a++){ cnt=0; for(b=a; b<=4; b++){ if (mun[b] == mun[a]) cnt=cnt+1; if(cnt>top) top=cnt; } printf("%d\n", top); } }</pre>	2

[프로그램 해설]

- a=0이면 b=0~4까지 변한다. mun[0]='h', mun[b]에서 'h'가 나오는 경우는 b=0일 경우만 해당된다. 따라서 cnt=1, cnt>top을 만족하므로 top=1이 된다.
- a=1이면 b=1~4까지 변한다. mun[1]='e', mun[b]에서 'e'가 나오는 경우는 b=1 일 경우만 해당된다. top의 값에는 변화가 없다.
- a=2이면 b=2~4까지 변한다. mun[2]='l', mun[b]에서 'l'이 나오는 경우는 b=2, 3인 경우 두 가지이므로 cnt=2, 그러면 cnt>top을 만족하므로 top=2가 된다.
- 같은 방법으로 a=3인 경우와 a=4인 경우를 분석하면 top의 값에는 변화가 없다.
- 따라서 top의 최종값은 2가 된다.



1

2009학년도 대수능

다음 프로그램의 실행 결과는?

```
#include <stdio.h>
void main(){
    int a, b, f[5];
    int d[5] = {35, 54, 59, 61, 47};
    for(a=0; a<=4; a++) {
        f[a] = d[a] / 10;
    }
    for(a=0; a<=4; a++) {
        for(b=0; b<=f[a]-1; b++)
            printf("■");
        printf("\n");
    }
}
```

- ①

③

⑤

②

④

2

2010학년도 대수능

다음 프로그램의 실행 결과는?

```
#include <stdio.h>
void main() {
    int a, b;
    int c[] = {0, 0, 0, 0};
    int d[] = {1, 2, 3, 1, 4};
    for(a=0; a<=4; a++) {
        c[d[a]-1] = c[d[a]-1] + 1;
    }
    for(a=0; a<=3; a++) {
        for(b=1; b<=c[a]; b++) {
            printf("* ");
        }
        printf("\n");
    }
}
```

- ①

③

⑤

②

④

3 다음 프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>
void main(){
    int a, sum = 0;
    int num[6] = { 5, 3, 6, 4, 15, 7};
    for(a = 0; a <= 5; a++)
        if (num[a] % 2 == 0)
            sum = sum + num[a];
    printf("%2d\n", sum);
}
```

- ① 10 ② 12 ③ 15
④ 17 ⑤ 23

4 다음 프로그램의 실행 결과 출력되는 값은?

```
#include <stdio.h>
void main(){
    int su[7] = {1,1,0,0,1,0,1};
    int a, cnt, pa;
    cnt = 0;
    pa = 0;
    for(a = 0; a <= 6; a++){
        if (su[a] == 1)
            cnt = cnt + 1;
    }
    if(cnt % 2 == 0)
        pa = 1;
    printf("%2d", pa);
}
```

- ① 0 ② 1 ③ 2
④ 3 ⑤ 4

5 다음 프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>
void main(){
    int k, m, count, sum;
    int num[7] = {2, 3, 7, 4, 3, 2, 4};

    count = 0, sum = 0;
    for(k = 0; k <= 5; k++) {
        for(m = k + 1; m <= 6; m++)
            if (num[k] == num[m]) {
                count = count + 1;
                sum = sum + num[m];
            }
    }
    printf("%2d %2d", count, sum);
}
```

- ① 3 9 ② 3 14 ③ 4 14
④ 4 16 ⑤ 5 18

6 다음 프로그램의 출력 결과는?

```
#include <stdio.h>
void main(){
    int k, m;
    int num[8] = {12, 43, 21, 9, 16, 8, 7, 5};
    int hash[5] = {0,0,0,0,0};
    for(k = 0; k <= 7; k++){
        m = num[k] % 5;
        hash[m] = hash[m] + 1;
    }
    for(k = 0; k <= 4; k++)
        printf("%2d", hash[k]);
}
```

- ① 10001 ② 12111 ③ 12221
④ 21112 ⑤ 22211

7 다음 프로그램의 실행 결과는?

```
#include <stdio.h>
void main(){
    int a, k, m;
    char t1, t2;
    int n[5]={3, 7, 1, 8, 9};
    char p[5] = {'A', 'B', 'C', 'D', 'E'};

    k = n[0], m = n[0];
    for(a = 0; a <= 4; a++){
        if(n[a] > k){
            k = n[a];
            t1 = p[a];
        }
        if (n[a] < m){
            m = n[a];
            t2 = p[a];
        }
    }
    printf("%c%c\n", t1, t2);
}
```

- ① AC ② BC ③ DE
④ EC ⑤ ED

8 다음 프로그램의 실행 결과는?

```
#include <stdio.h>
void main(){
    int k;
    int a[5] = {1, 5, 2, 3, 4};
    int b[5] = {2, 5, 2, 4, 4};

    for(k = 0; k <= 4; k++){
        if (a[k] == b[k])
            printf("@");
        else
            printf("&");
    }
}
```

- ① @@@ ② &&& ③ &@@@&
④ &@@@&@ ⑤ &&@&@

9 다음 프로그램의 출력 결과는?

```
#include <stdio.h>
void main(){
    int a, k=0;
    int n[5] = {1, 3, 2, 7, 6};

    for(a = 0; a <= 3; a++){
        if (n[a] > n[a+1])
            k = k + n[a];
    }
    printf("%d\n", k);
}
```

- ① 7 ② 8 ③ 9
④ 10 ⑤ 11