

12. 배열

■ 배열이란?

- 배열(array) : 형식(자료형, type)이 같은 자료 여러 개가 모여 새로운 하나를 이룬 형식
- 배열은 일반적인 변수와 달리 여러 개의 값을 저장할 수 있음. 똑같은 자료형(int, char, float..)을 모아놓은 변수라고 할 수 있음.
- 여러 개의 변수가 모여서 배열을 이루기 때문임.
- 1차원 배열의 선언

<pre>int nArr[5];</pre>	<ul style="list-style-type: none">● <code>int</code> : 배열을 이루는 요소(변수)의 자료형● <code>nArr</code> : 배열의 이름● <code>[5]</code> : 배열의 길이															
<p><code>int</code> 형 변수 5개로 이뤄진 배열을 선언, 그 배열의 이름은 <code>nArr</code>임!</p> <p><code>int</code> 형 변수 5개가 나란히 선언되어 있음</p> <p>배열 <code>nArr</code>의 자료형은? <code>int[5]</code></p> <div><div><div>nArr</div><table><tr><td>nArr[0]</td><td>nArr[1]</td><td>nArr[2]</td><td>nArr[3]</td><td>nArr[4]</td></tr><tr><td>int</td><td>int</td><td>int</td><td>int</td><td>int</td></tr><tr><td>4byte</td><td>4byte</td><td>4byte</td><td>4byte</td><td>4byte</td></tr></table></div></div>		nArr[0]	nArr[1]	nArr[2]	nArr[3]	nArr[4]	int	int	int	int	int	4byte	4byte	4byte	4byte	4byte
nArr[0]	nArr[1]	nArr[2]	nArr[3]	nArr[4]												
int	int	int	int	int												
4byte	4byte	4byte	4byte	4byte												

- 1차원 배열의 접근

<pre>int nArr[5]; nArr[0] = 12; // 배열 nArr의 첫 번째 요소에 12를 저장해라! nArr[1] = 34; // 배열 nArr의 두 번째 요소에 34를 저장해라! nArr[2] = 56; // 배열 nArr의 세 번째 요소에 56을 저장해라! nArr[3] = 78; // 배열 nArr의 네 번째 요소에 78을 저장해라! nArr[4] = 90; // 배열 nArr의 다섯 번째 요소에 90을 저장해라!</pre>
--

- [] 연산자 사이에 숫자는 배열의 위치 정보를 알려준다.(인덱스)
- 배열의 위치 정보를 명시하는 인덱스 값은 1이 아닌 **0부터 시작**한다.
- 배열 예제(1)

<pre>#include<stdio.h> int main(){ int nArr[5]; int i, sum = 0; nArr[0] = 10, nArr[1] = 20, nArr[2] = 30, nArr[3] = 40, nArr[4] = 50; for (i = 0; i < 5; i++) { printf("nArr[%d] = %d\n", i, nArr[i]); } printf("배열 요소에 저장된 값의 합 : %d\n", sum); return 0; }</pre>	<div>실행결과</div>
---	-----------------

■ 배열 선언과 동시에 초기화 하기

- 1차원 배열의 선언과 초기화 방법

```
int nArr1[5] = { 1,2,3,4,5 };
```

nArr1	0x00d5fcb8 {1, 2, 3, 4, 5}
[0]	1
[1]	2
[2]	3
[3]	4
[4]	5

```
int nArr2[] = { 2,4,6,8,10,12,14,16 };
```

nArr2	0x00d5fc90 {2, 4, 6, 8, 10, 12, 14, 16}
[0]	2
[1]	4
[2]	6
[3]	8
[4]	10
[5]	12
[6]	14
[7]	16

```
int nArr3[10] = { 1,3 };
```

nArr3	0x00d5fc60 {1, 3, 0, 0, 0, 0, 0, 0, 0, 0}
[0]	1
[1]	3
[2]	0
[3]	0
[4]	0
[5]	0
[6]	0
[7]	0
[8]	0
[9]	0

```
int nArr3[]; //실행오류! 인덱스 값을 알 수 없음
```

배열 예제(2)

```
#include<stdio.h>
int main(){
    int nArr1[5] = { 10,20,30,40,50 };
    int nArr2[5] = { 0 };
    int i = 0;

    // 배열 전체 요소를 동시에 대입하려 시도한다.

    nArr2 = nArr1;

    for (i = 0; i < 5; i++) {
        printf(" nArr2[%d] : %d\n", i, nArr2[i]);
    }
    printf("\n");
    return 0;
}
```

- error C2106: '=': 왼쪽 피연산자는 l-value이어야 합니다.
- 단순 대입 연산자의 왼쪽 피 연산자가 변수가 아니어서 발생한 것임.
- 배열 이름의 실체는 '주소 상수'

● 배열 예제(3) - 위의 코드 에러 수정

```
#include<stdio.h>
int main()
{
    int nArr1[5] = { 10,20,30,40,50 };
    int nArr2[5] = { 0 };
    int i = 0;

    // nArr2 = nArr1; 와 같은 코드는 불가능
    // 아래와 같이 반복문으로 하나씩 복사해야 함.
    for (i = 0; i < 5; i++) {
        nArr2[i] = nArr1[i];
    }

    for (i = 0; i < 5; i++) {
        printf(" nArr2[%d] : %d\n", i, nArr2[i]);
    }
    printf("\n");
    return 0;
}
```

실행결과

Microsoft Visual S

```
nArr2[0] : 10
nArr2[1] : 20
nArr2[2] : 30
nArr2[3] : 40
nArr2[4] : 50
```

배열 예제(4)

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
int main()
{
    int nArr1[5], i;

    printf(" 배열의 요소 5개를 입력하세요.\n");
    for (i = 0; i < 5; i++) {
        printf(" nArr1[%d] : ", i);
        scanf("%d", &nArr1[i]);
    }

    printf("\n 배열을 출력합니다.\n");
    for (i = 0; i < 5; i++) {
        printf(" nArr1[%d] : %d\n", i, nArr1[i]);
    }
    printf("\n");
    return 0;
}
```

실행결과

Microsoft Visual Studio 디버그 콘솔

```
배열의 요소 5개를 입력하세요.
nArr1[0] : 9
nArr1[1] : 8
nArr1[2] : 5
nArr1[3] : 1
nArr1[4] : 6

배열을 출력합니다.
nArr1[0] : 9
nArr1[1] : 8
nArr1[2] : 5
nArr1[3] : 1
nArr1[4] : 6
```

16-1 [1차원 배열의 활용]

1. 길이가 5인 double형 배열을 선언해서 프로그램 사용자로부터 총 5개의 실수를 입력받자! 그리고 입력이 끝나면 배열 요소의 최대값, 최소값, 합계, 평균을 출력하도록 프로그램을 작성해보자!

예시)

Microsoft Visual Studio 디버그 콘솔

```
배열의 요소 5개를 입력하세요 : 1.23 -4.5678 9.01234 56.78901 23.45
배열의 요소 : 1.230000 -4.567800 9.012340 56.789010 23.450000
최대값: 56.789010, 최소값: -4.567800, 합계: 85.913550, 평균: 17.182710
```

■ 문자의 배열

- 문자열의 실체는 char형 배열
- 상수 형태로 기술하는 "Nice to meet you!" 과 같은 문자열도 사실은 배열임!

printf("%s", "Nice to meet you!")는 char str[18]의 배열에 저장된 모습과 같다.

```
char str[18] = "Nice to meet you!";
```

str	0x010ffa10 "Nice to meet you!"
[0]	78 'N'
[1]	105 'i'
[2]	99 'c'
[3]	101 'e'
[4]	32 ' '
[5]	116 't'
[6]	111 'o'
[7]	32 ' '
[8]	109 'm'
[9]	101 'e'
[10]	101 'e'
[11]	116 't'
[12]	32 ' '
[13]	121 'y'
[14]	111 'o'
[15]	117 'u'
[16]	33 '!'
[17]	0 '\0'

또 다른 초기화 방법

```
char str2[] = { 'G','o','o','d',' ','t','o',' ','s','e','e',' ','y','o','u',' ','a','g','a','i','n','!','\0' };
```

```
char str2[] = { "Good to see you again!" };
```

str2	0x010ffda0 "Good to see you again!"
[0]	71 'G'
[1]	111 'o'
[2]	111 'o'
[3]	100 'd'
[4]	32 ' '
[5]	116 't'
[6]	111 'o'
[7]	32 ' '
[8]	115 's'
[9]	101 'e'
[10]	101 'e'
[11]	32 ' '
[12]	121 'y'
[13]	111 'o'
[14]	117 'u'
[15]	32 ' '
[16]	97 'a'
[17]	103 'g'
[18]	97 'a'
[19]	105 'i'
[20]	110 'n'
[21]	33 '!'
[22]	0 '\0'

문자열 예제(1)

```
#include<stdio.h>
int main()
{
    char str[18] = "Nice to meet you!";
    printf("배열 str의 크기 : %d\n", sizeof(str));
    //sizeof함수는 자료형이나 변수의 크기를 바이트 단위로 계산해줌
    printf("NULL 문자 문자형 출력 : %c\n", str[17]);
    printf("NULL 문자 정수형 출력 : %d\n", str[17]);

    str[16] = '?';
    printf("문자열 출력 : %s\n", str);
    return 0;
}
```

실행결과

Microsoft Visual Studio 디버그 콘솔

```
배열 str의 크기 : 18
NULL 문자 문자형 출력 :
NULL 문자 정수형 출력 : 0
문자열 출력 : Nice to meet you?
```

- 문자형태로 출력할 경우 널(NULL)문자이며, 널은 아무런 출력이 발생하지 않음.
- 널(NULL)문자의 아스키 코드 값은 0이다.

● 널 문자와 공백문자 확인

```
#include<stdio.h>
int main()
{
    char nu = '\0'; // 널 문자 저장
    char sp = ' '; // 공백 문자 저장
    printf("%d %d\n", nu, sp);
    return 0;
}
```

- 공백 문자의 아스키 코드값은 32로, 널 문자의 아스키 코드값으로 0으로 서로 다르다!

1. 5개의 정수를 입력받아 그대로 출력하는 프로그램을 작성하시오.(배열로 풀 것)

<p>입력 예시1</p> <p>1 2 3 4 5</p>	<p>입력 예시2</p> <p>5 6 9 8 3</p>
<p>출력 예시1</p> <p>Microsoft Visual Studio 디버그 콘솔</p> <pre>1 2 3 4 5 1 2 3 4 5</pre>	<p>출력 예시2</p> <p>Microsoft Visual Studio 디버그 콘솔</p> <pre>5 6 9 8 3 5 6 9 8 3</pre>

2. 출력 결과를 보고 아래 빈칸을 채워보세요.

```
#include<stdio.h>
int main(){
    int num[10];
    for (int i = 0; i < 10; i++) {
        // num 배열에 1부터 10까지의 숫자 저장하는 코드 작성
    }
    // 출력
    return 0;
}
```

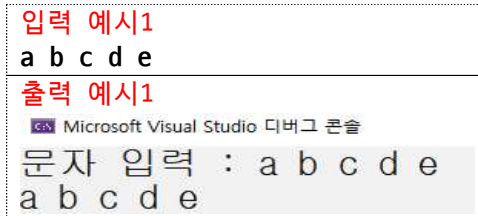
실행결과

Microsoft Visual Studio 디버그 콘솔

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

3. 5개의 문자를 입력받아 그대로 출력하는 프로그램을 작성하시오.(배열로 풀 것)

+) 버퍼 공간을 지우기 위해 scanf를 할 때 공백을 앞에 넣는다.



4. 출력 결과를 보고 아래 빈칸을 채워보세요.

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
int main()
{
    int cnt = 0; // 이름의 길이를 세는 변수
    char sName[100], sNum[10]; // 이름과 학번을 문자열로 저장하는 배열
    printf("이름을 영어로 입력하세요 : ");
    scanf("%s", sName); // 이름을 문자열로 입력받기
    printf("학번을 입력하세요 : ");
    // 학번 입력받기

    while (sName[cnt] != '\0')
    {
        // 이름의 길이 증가시키기
    }

    // 마지막 출력 결과처럼 나오게 하기
    return 0;
}
```

실행결과

Microsoft Visual Studio 디버그 콘솔
이름을 영어로 입력하세요 : eunjin
학번을 입력하세요 : 1400
당신의 이름은 eunjin(6글자)이고 학번은 1400입니다.

● 문자열 예제(3)

```
#include<stdio.h>
int main()
{
    char str[50] = "I Like C Programming!!";
    printf("string : %s\n", str);

    str[8] = '\0'; // 인덱스 8번 널 문자로 저장
    printf("string : %s\n", str);

    str[6] = '\0'; // 인덱스 6번 널 문자로 저장
    printf("string : %s\n", str);

    str[1] = 0; // 인덱스 1번 문자를 널 문자로 저장(아스키 코드로 널문자 저장)
    printf("string : %s\n", str);

    return 0;
}
```

실행결과

Microsoft Visual Studio 디버그 콘솔
string : I Like C Programming!!
string : I Like C
string : I Like
string : I

● 문자열을 출력하면 널문자 '\0' 전까지 출력이 된다.

1. 10개의 정수를 입력받아 수의 합계와 평균을 구하는 프로그램을 작성하시오.

출력 예시

Microsoft Visual Studio 디버그 콘솔

```
10 20 30 40 50 60 70 80 90 100
입력한 수의 합계 : 550
입력한 수의 평균 : 55
```

Microsoft Visual Studio 디버그 콘솔

```
56 98 61 52 37 59 89 99 60 90
입력한 수의 합계 : 701
입력한 수의 평균 : 70
```

2. 충분한 크기의 정수를 저장하는 배열을 만들고 계속 입력받다가 0이 입력되면 0을 제외한 그 때까지 입력된 정수를 뒤에서부터 출력하는 프로그램을 작성하시오.

출력 예시

Microsoft Visual Studio 디버그 콘솔

```
26 55 33 62 88 9 -8 6 0
6 -8 9 88 62 33 55 26
```

Microsoft Visual Studio 디버그 콘솔

```
3 2 1 1 9 0
9 1 1 2 3
```

Microsoft Visual Studio 디버그 콘솔

```
0
```

3. 10개의 자연수를 입력받아 가장 큰 수를 출력하는 프로그램을 작성하시오. (배열로)

출력 예시

Microsoft Visual Studio 디버그 콘솔

```
5 10 8 55 6 31 12 24 61 2
61
```

Microsoft Visual Studio 디버그 콘솔

```
55 99 63 2 0 4 5 77 150 6
150
```

4. 충분한 크기의 배열을 선언하여 계속 입력받다가 1000 이상의 숫자가 입력되면 반복을 멈추고 지금까지 입력한 숫자 중 가장 작은 숫자를 출력하는 프로그램을 작성하시오.

출력 예시

Microsoft Visual Studio 디버그 콘솔

```
5 9 6 2 50 999 600 1000
2
```

Microsoft Visual Studio 디버그 콘솔

```
500 9 600 4 80 99 55 63 48 22 69 7 5 3 4 9462
3
```

5. 충분한 크기의 배열을 선언하여 계속 입력받다가 1000 이상의 숫자가 입력되면 반복을 멈추고 지금까지 입력한 숫자 중 가장 작은 숫자를 출력하는 프로그램을 작성하시오.

출력 예시

Microsoft Visual Studio 디버그 콘솔

```
5 9 6 2 50 999 600 1000
2
```


Microsoft Visual Studio 디버그 콘솔


```
500 9 600 4 80 99 55 63 48 22 69 7 5 3 4 9462
3
```

연습 문제 (1차원 배열)

 **브론즈 III** <https://www.acmicpc.net/problem/10818> 최소, 최대

 **브론즈 II** <https://www.acmicpc.net/problem/3052> 나머지

 **브론즈 II** <https://www.acmicpc.net/problem/8958> OX퀴즈

 **브론즈 I** <https://www.acmicpc.net/problem/1546> 평균

 **브론즈 I** <https://www.acmicpc.net/problem/4344> 평균은 넘겠지