초급 C언어

가천대학교
Gachon University


배열(Array)

1차원 배열-숫자(1)

- 배열의 선언

형식) 자료형　배열명[배열 크기];

예)　int s[10];

→ 10개의 int형 값을 가지는 변수의 모임인 배열 s

※ int : 배열 원소들이 갖는 자료형

s　: 배열 이름

10 : 배열 원소의 개수로 배열 크기

s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]
int형	int형	int형	int형	int형	int형	int형	int형	int형	int형

1차원 배열-숫자(2)

- 배열에 값을 넣고 출력하기(1) : 배열명과 인덱스를 사용

예)

```
int s[10];
int i=8;


s[5] = 80;
s[1] = s[5];
s[i] = 10;
…
scanf_s("%d", &s[i]); //키보드에서 32 입력을 가정
printf("s[%d] :%d₩n",i, s[i]);
```

s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]
	80				80			~~10~~ 32	

1차원 배열-숫자(3)

• 배열에 값을 넣고 출력하기(2) : 배열 선언시 값을 넣는 초기화와 반복문을 사용

형식)

- 배열 초기화 : 배열 선언시 =와 { }를 사용하여 { }내부에 배열 원소별 값을 첨자 순서대로 나열

예) double f[3]={1.1, 2.2, 3.3} ;

f[0]	f[1]	f[2]
1.1	2.2	3.3
double형	double형	double형

- [ ]내의 숫자인 인덱스가 0에서 (배열 크기-1)로 변경되는 점을 활용하여 반복문으로 배열을 사용

예)

```
for (i=0; i<3; i++ ) {
        printf("f[%d]: %lf₩n", i, f[i]); // 1.100000  2.200000 3.300000
}
```

→ 데이터를 배열과 반복문으로 관리하는 것은 컴퓨터로 대량의 데이터를 처리해야 하는 실무에서 대단히 유용하다.

[실습] 다음 조건을 만족하는 프로그램을 작성하시오.

[결과]
소프트웨어 교과의 점수를 입력하세요. (5명)
==========
[1]번 :***80***
[2]번 :***90***
[3]번 :***70***
[4]번 :***60***
[5]번 :***100***

소프트웨어 교과의 평균점수 : 80.000000

[조건]
- 1개 학급의 소프트웨어 개론 교과의 평균 점수를 계산하여 출력
- 1개 학급은 5명의 학생으로 구성되어 있음을 가정
  ※ 실제 환경에서 학급당 인원수는 많아야 하나, 프로그래밍 연습의 편의를 위해 5명으로 제한
- 5명의 중간고사 성적은 입력받음(ex. scanf_s("%d", &s[i]);)
- 배열을 사용

문자열과 문자 배열(1)

- 문자열 상수를 저장하기 위해서 문자 배열을 사용한다.

예) char str[4], char city[10] 등

- 문자 배열의 크기는 '₩0'을 저장하기 위해 문자열 상수에 포함된 문자 수보다 최소한 한 개가 더 많아야 한다.

예) char str[3+1], char city[9+1] 등

문자열과 문자 배열(2)

- scanf_s로 문자열을 입력하는 방법

  -%s 형식을 사용

  -배열명 앞에 &를 사용하지 않으며, 문자 배열의 공간 수를 알려주어야 함

  - 메모리에 저장되는 변수의 공간 수를 알려주는 sizeof를 사용할 수 있음

  예) char city[10];

  scanf_s("%s", city, 10);

  // scanf_s("%s", city, sizeof(city));와 동일한 의미

  -문자열 다음에 널 문자를 자동으로 추가함

- printf로 문자열을 출력하는 방법

  1) %s 형식과 배열명을 사용 : 배열의 처음 원소부터 '₩0'을 만날 때 까지의 값을 출력

  예) char city[10];

  printf("%s", city);

  2) %c 형식과 반복문을 사용 : 반복문에서 해당 인덱스의 배열 값을 출력

  예) char city[10];

  for(i=0; i <10; i++){

  printf("%c", city[i]);

  }

sizeof 연산자

- **sizeof** 연산자 : 주어진 값이나 변수 또는 자료형이 메모리 공간에서 차지하는 크기를 바이트 단위로 알려 줌

- 형식) **sizeof(**값이나 변수 또는 자료형**);**

예)

✓ sizeof(char) : 1
✓ sizeof(short) : 2 sizeof(int) : 4
✓ sizeof(float) : 4 sizeof(double) : 8

※ 주의) 자료형의 경우 일반적으로 위와 같은 결과가 나오지만, 시스템에 따라 자료형이 메모리 공간에서 차지하는 크기가 다를 수 있으므로, 필요한 경우 sizeof 연산자를 사용하여 확인한 후 사용!!!

✓ sizeof("Hello") : 6 //영어는 1글자에 1바이트를 차지 + 널문자 1바이트
✓ sizeof("김가천") : 7 //한글은 1글자에 2바이트를 차지 + 널문자 1바이트

문자 배열의 초기화

방법 1) { }내부에 널문자를 포함하여 한 문자씩 ,로 구분하여 초기화할 수 있다.

※ 정해진 배열 크기보다 적은 수의 초기값을 설정하면, 나머지 배열 원소에는 '₩0'이 저장된다.

예) char str[4] = { ' a ' , ' b ' , ' ₩0 ' };

str[0]	str[1]	str[2]	str[3]
a	b	'₩0'	'₩0'

- 방법 2) "..."로 표시된 문자열로 초기화할 수 있다.

예) char str[4] = "abc";

str[0]	str[1]	str[2]	str[3]
a	b	c	'₩0'

※ 문자 배열의 크기를 쓰지 않고 "..."로 초기화 하면, (초기화 값의 개수 +1) 크기의 문자 배열이 만들어 진다.

예) char str[] = "abcd";

str[0]	str[1]	str[2]	str[3]	str[4]
a	b	c	d	'₩0'

[예제] 다음 프로그램의 결과를 예측하여 보고 비교하시오.

```
#include <stdio.h>
int main(){
   char name[10];

   printf("이름을 입력하세요 : ");
   scanf_s("%s", name, sizeof(name));
   printf("%s₩n", name);

   return 0;
}
```

[조건]

- 첫번째 실행시 입력
  이름을 입력하세요 : ***황혜정***
- 두번째 실행시 입력
  이름을 입력하세요 : ***황 혜정***

[결과] 다음 프로그램의 결과를 예측하여 보고 비교하시오.

[결과]

- 첫번째 실행시 입력

  이름을 입력하세요 : ***황혜정***

  황혜정

- 두번째 실행시 입력

  이름을 입력하세요 : ***황 혜정***

  황

- scanf_s는 문자열 자료의 입력으로 공백 전까지의 자료를 받는다.
  예) '황 혜정'을 입력하면, "황"만 배열에 저장

- 공백을 포함한 문자형 자료를 입력 받으려면, gets_s나 getchar와 같은 라이브러리 함수를 사용

[실습] 다음 조건을 만족하는 프로그램을 작성하시오.

[결과]
문자열 C language의 길이는 10입니다.
C language의 역순 : egaugnal C

[조건]
- 반복문을 사용하여 주어진 문자열의 길이 구하기
  - 문자열 저장하는 배열의 크기는 15로 하기
  - 반복문에서 조건식에 '₩0'를 사용
- 구한 길이와 반복문을 사용하여 문자열을 역순으로 출력

2차원 배열

- 2차원 배열
  - 1차원 배열이 여러 개 모인 배열
  - 표 형태의 구조로 표현되는 자료들을 저장하는 데 사용

  - 표의 가로 단위를 '행'이라고 하며,
  - 행을 나눈 단위를 '열 ' 이라고 한다.

- 형식) 자료형 배열명[행의 개수][열의 개수]

예) int s[3][5];

→각 행은 첫번째 [ ]속에 0부터 시작하여 (행의 개수-1)까지의 인덱스로 표현

→각 행의 열은 두번째 [ ]속에 0부터 시작하여 (열의 개수-1)까지의 인덱스로 표현

5열

3행					
	s[0][0]	s[0][1]	s[0][2]	s[0][3]	s[0][4]
	s[1][0]	s[1][1]	s[1][2]	s[1][3]	s[1][4]
	s[2][0]	s[2][1]	s[2][2]	s[2][3]	s[2][4]

2차원 배열의 초기화 - 숫자

- 각 행의 열의 값을 { }안에 ,로 구분하여 써 준다.
- 모든 행을 { }안에 ,로 구분하여 써 준다.

예)

```
int s[3][5]
= {
     { 0,  1,  2,  3,  4 },    // 첫 번째 행의 5개 열의 초기값
     { 10, 11, 12, 13, 14 }, // 두 번째 행의 5개 열의 초기값
     { 20, 21, 22, 23, 24 }  // 세 번째 행의 5개 열의 초기값
   };
```

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

2차원 배열의 입력 및 출력 방법

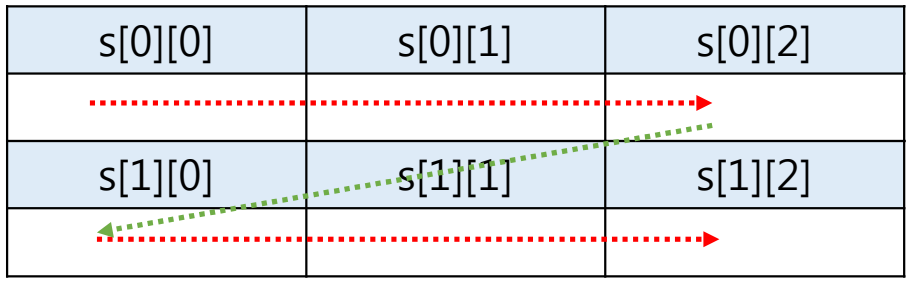
- 중첩 반복문을 사용하여 각 행에 대한 열에 접근하여 모든 배열 값을 입출력 할 수 있다.

예)

```
int s[2][3];

for (i=0; i<2; i++) {  // 행 단위
   for (j=0; j<3; j++) {  //열이 바뀜
      printf("%d행 %d열? ", i, j);
      scanf_s("%d", &s[i][j]);
  }
}

for (i=0; i<2; i++) {  // 행 단위
   for (j=0; j<3; j++) {  //열이 바뀜
      printf("%d ", s[i][j]);
  }
}
...
```



char형 2차원 배열과 초기화(1)

- 7글자 이하의 4개 도시명을 저장

  char city[4][8];

  - '행'의 의미: 배열에 저장할 문자열의 개수(4개 도시명)
  - '열'의 의미: 하나의 문자열에 널을 포함한 최대 문자수(7글자 이하 도시명 + 1)

- 초기값 지정: { }속에 ,로 구분된 각 행 단위별 문자열을 ""로 써준다.

  예) char city[4][7+1]={"Seoul", "Busan", "Incheon", "Ulsan"};

city[0][0]	city[0][1]	city[0][2]	city[0][3]	city[0][4]	city[0][5]	city[0][6]	city[0][7]
S	e	o	u	l	₩0		
city[1][0]	city[1][1]	city[1][2]	city[1][3]	city[1][4]	city[1][5]	city[1][6]	city[1][7]
B	u	s	a	n	₩0		
city[2][0]	city[2][1]	city[2][2]	city[2][3]	city[2][4]	city[2][5]	city[2][6]	city[2][7]
I	n	c	h	e	o	n	₩0
city[3][0]	city[3][1]	city[3][2]	city[3][3]	city[3][4]	city[3][5]	city[3][6]	city[3][7]
U	l	s	a	n	₩0		

char형 2차원 배열과 초기화(2)

- 7글자 이하의 4개 도시명을 저장
  char city[4][8]={"Seoul", "Busan", "Incheon", "Ulsan"};

  - 문자열 배열에 인덱스를 하나만 사용하면 배열에 보관된 문자열에 접근한다.
  예) city[0]는 "Seoul"을 의미, city[3]은 "Ulsan"을 의미

  - 문자열 배열에 인덱스를 두 개 사용하면 특정 문자열의 특정 위치에 있는 문자에 접근한다.
  예) city[0][0]는 S를 의미, city[1][0]은 B를 의미

city[0] →	city[0][0]	city[0][1]	city[0][2]	city[0][3]	city[0][4]	city[0][5]	city[0][6]	city[0][7]
	S	e	o	u	l	₩0		
city[1] →	city[1][0]	city[1][1]	city[1][2]	city[1][3]	city[1][4]	city[1][5]	city[1][6]	city[1][7]
	B	u	s	a	n	₩0		
city[2] →	city[2][0]	city[2][1]	city[2][2]	city[2][3]	city[2][4]	city[2][5]	city[2][6]	city[2][7]
	I	n	c	h	e	o	n	₩0
city[3] →	city[3][0]	city[3][1]	city[3][2]	city[3][3]	city[3][4]	city[3][5]	city[3][6]	city[3][7]
	U	l	s	a	n	₩0		

char형 2차원 배열의 입력과 출력(1)

char city[4][8]={"Seoul", "Busan", "Incheon", "Ulsan"};

- scanf_s 사용시

  - Seoul, Busan과 같이 각 행 단위별 문자열을 나타내는 city[0], city[1]을 &를 붙이지 않고, %s와 반복문으로 입력받음

  예)

```
for (i = 0; i < 4; i++) {
    scanf_s("%s", city[i], sizeof(city[i]));
}
```

city[0] →	city[0][0]	city[0][1]	city[0][2]	city[0][3]	city[0][4]	city[0][5]	city[0][6]	city[0][7]
city[1] →	city[1][0]	city[1][1]	city[1][2]	city[1][3]	city[1][4]	city[1][5]	city[1][6]	city[1][7]
city[2] →	city[2][0]	city[2][1]	city[2][2]	city[2][3]	city[2][4]	city[2][5]	city[2][6]	city[2][7]
city[3] →	city[3][0]	city[3][1]	city[3][2]	city[3][3]	city[3][4]	city[3][5]	city[3][6]	city[3][7]

char형 2차원 배열의 입력과 출력(2)

char city[4][8]={"Seoul", "Busan", "Incheon", "Ulsan"};

- printf 사용시

1) Seoul, Busan과 같이 각 행 단위별 문자열을 나타내는 city[0], city[1]를 %s와 반복문으로 출력

예)

```
for (i = 0; i < 4; i++) {
      printf("%s ", city[i]);
}
```

2) 각 행 단위별 문자를 나타내는 city[0][0], city[0][1]를 %c와 중첩반복문으로 출력

예)

```
for (i = 0; i < 4; i++) {
   for (j = 0; j < 8; j++) {
      printf("%c", city[i][j]);
   }
   printf("₩n");
}
```

[실습] 다음과 같은 결과가 출력되도록 코딩하시오.

[결과]

7자 이하 영어 도시명을 4개 입력하세요.

Seoul

Busan

Incheon

Ulsan

입력한 도시명

Seoul Busan Incheon Ulsan

Seoul

Busan

Incheon

Ulsan

[조건]

- 입력시 문자열 단위로 받음
- 출력시 문자단위와 문자열 단위로 출력
  - 4개의 도시를 구분할 때는 For문을 사용
  - 각 도시 이름을 출력할 때는 조건식에 '₩0'를 사용한 while문을 사용

매크로 상수

- 배열의 크기와 같이 여러 곳에서 반복 사용되는 상수를 수정해야 할 때 한번에 변경할 수 있도록 매크로 상수를 사용할 수 있다.

- 형식) #define 매크로명 값

예) #define MAX 100

- 사용 방식)
- 프로그램 내에서 매크로명이 나오는 곳은 모두 해당 값으로 변경되어 컴파일된다.

예) printf("%d₩n", MAX);는 printf("%d₩n", 100);으로 컴파일됨

- 단, 문자열을 나타내는 "..." 내에서는 해당 값으로 변경되지 않는다.

예) printf("MAX : %d₩n", MAX);는 printf("MAX : %d₩n", 100);으로 컴파일됨

- 매크로 명은 일반적으로 대문자를 사용한다.
- #define 문장은 일반적으로 #include문과 int main{...} 사이에 작성한다.

[실습] 다음과 같은 결과가 나오도록 프로그램을 작성하시오.

[결과]

```
========================================
      학생명       성적
========================================
       김가천         80
       홍길동         90
       홍길순         70
       전지현         60
       남상미        100
========================================
소프트웨어 교과의 평균점수 : 80.00
========================================
```

[조건]

- 5명의 학생으로 구성되어 있는 1개 학급의 소프트웨어 개론 교과에 대해 다음을 처리
  - 개인별 이름과 해당 성적은 프로그램 내부에 임의의 값으로 저장되어 있음을 가정하며 그 내용을 출력
  - 소프트웨어 개론 교과의 평균 점수를 소수이하 3자리에서 반올림하여 출력
- 성적은 정수형 1차원 배열을 사용하며, 이름은 문자형 2차원 배열을 사용
  - score : 성적을 저장하는 배열명칭
  - name : 이름을 저장하는 배열명칭
- 배열 크기를 매크로 상수로 표현

```
#define SIZE_1 5  //학생수
#define SIZE_2 7  //이름을 저장하는 문자열의 길이
```

T h a n k y o u


TECHNOLOGY
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Velit ex
Incabo ipsum. labore sed tempora ratione asperiores des
labore sed tempora rati [gert one bore sed temi