

# Chapter 7. Arrays

May, 2016

Seungjae Baek

Dept. of software  
Dankook University

<http://embedded.dankook.ac.kr/~baeksj>

- 배열의 정의를 이해한다.
- 배열의 선언 방법을 이해한다.
- 각 배열 원소를 접근하는 방법을 이해한다.
- 문자열의 특징을 이해한다.
- 문자열 관련 라이브러리의 사용 방법을 이해한다.
- 다차원 배열을 이해한다.
- 배열의 초기화 방법을 이해한다.
- 문자열의 배열을 이해한다.
- 이 장의 결론

# 1차원 배열의 선언 (1/10)

- 배열이란 : 같은 유형을 갖는 변수들의 집합
- 형식

```
type variable_name[size];
```

```
#include <stdio.h> // array example

int main()
{
    int i;
    int a[5]; // array define

    for (i=0; i<5; i++)
        a[i] = i*2;

    for (i=0; i<5; i++)
        printf("%d\n", a[i]);

    return 0;
}
```

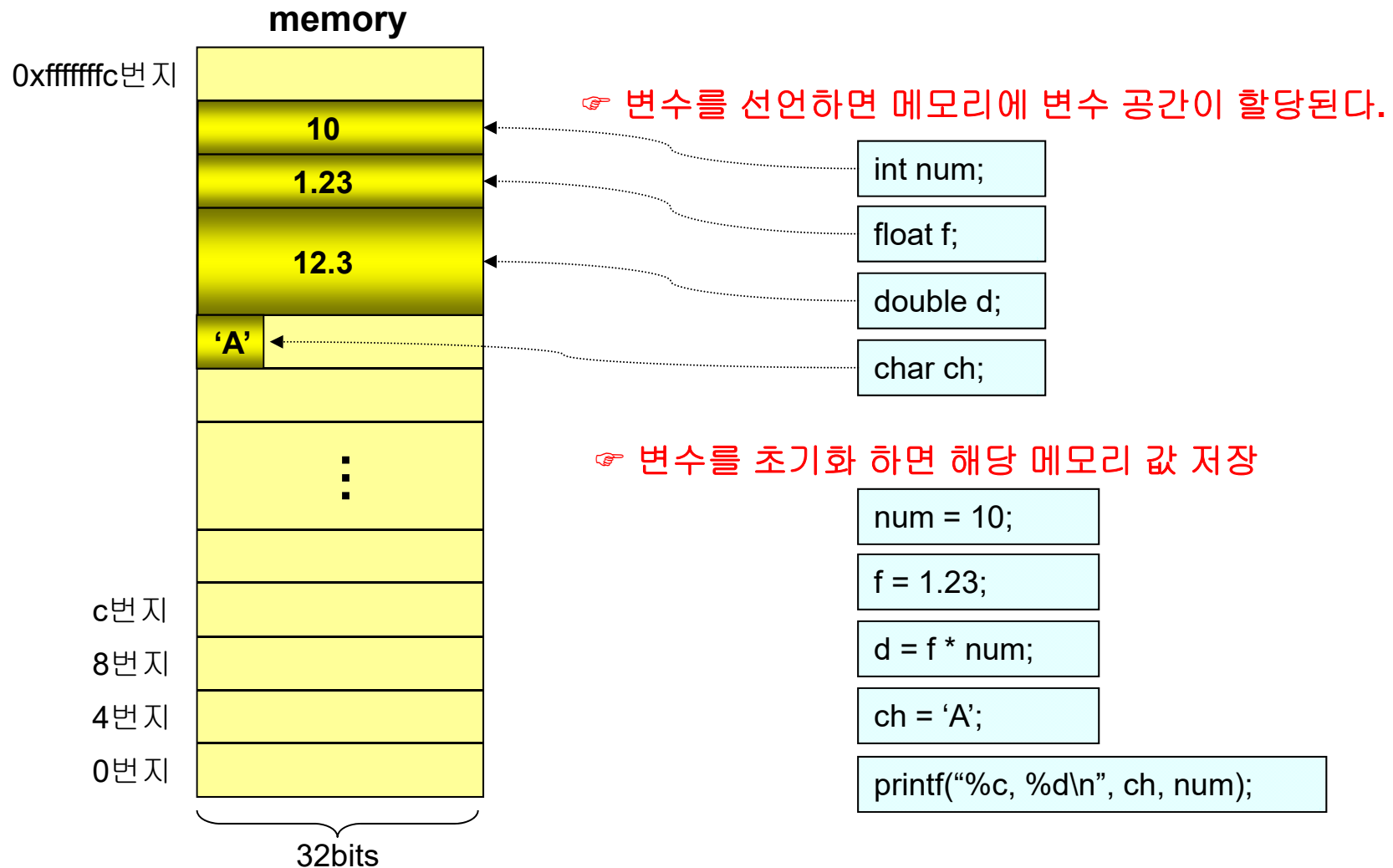
a를 배열의 이름(name)이라고 함

a[0], a[1], a[2], a[3], a[4]를 각각 배열의 원소(element)라고 함

배열에서 색인(index)는 0부터 size-1까지 사용가능

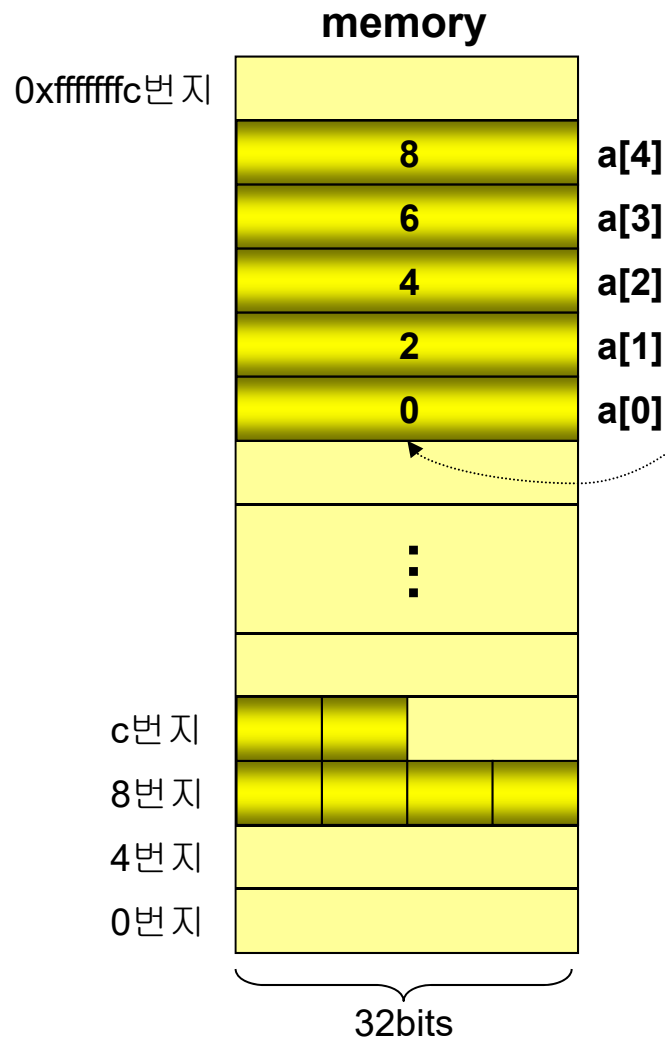
# 1차원 배열의 선언 (2/10)

## ■ C의 기본 유형 변수 선언 및 접근



# 1차원 배열의 선언 (3/10)

## ■ 배열 선언 및 각 원소 접근



### ☞ 배열을 선언하면

```
int a[5];
```

메모리 입장에서  
**int a[5];**는 **int v,w,x,y,z;**를 선언하고  
**a[0], a[1], a[2], a[3], a[4]** 대신  
**v, w, x, y, z**로 접근하는 것과 동일.  
**a**는 배열의 시작 위치, 즉 시작하는 메모리 주소를 가리킴 (**a == &a[0]**,  
**주의: a**는 포인터 상수임!!)

### ☞ 배열의 각 원소 접근

```
for (i=0; i<5; i++) a[i] = i*2;  
printf("%d", a[3]);
```

### ☞ 문자형 배열을 선언하면

```
char b[6];
```

# 1차원 배열의 선언 (4/10)

## ■ 배열의 장점

✓ 배열을 이용한 지난 과제 재 구현

```
#include <stdio.h> // 배열 사용하지 않음
#include <stdlib.h>
#include <time.h>

int main()
{
    int num0, num1, num2, num3, num4, num5;
```

```
#include <stdio.h> // 배열 사용
#include <stdlib.h>
#include <time.h>

int main()
{
    int num[6], i, j;
```

```
    srand(time(NULL));

    while (1) {
        num0 = rand()%45 + 1;
        num1 = rand()%45 + 1;
        num2 = rand()%45 + 1;
        num3 = rand()%45 + 1;
        num4 = rand()%45 + 1;
        num5 = rand()%45 + 1;

        if ((num0!=num1) && (num0!=num2) && (num0!=num3) &&
            (num0!=num4) && (num1!=num2) && (num1!=num3) &&
            (num1!=num4) && (num2!=num3) && (num2!=num4) &&
            (num3!=num4))
            break;
    }
    printf("%d,%d,%d,%d,%d,%d\n",
           num0,num1,num2,num3,num4, num5);
}
```

```
    srand(time(NULL));

    for (i=0; i<6; i++) {
        num[i] = rand()%45 + 1;

        for (j=0; j<i; j++) {
            if (num[j] == num[i]) {
                i--;
                break;
            }
        }
    }

    for (i=0; i<6; i++)
        printf("%d ", num[i]);
}
```

서로 다른 숫자 60개라면?

☞ 문제 풀이 방법(**algorithm**)이 결정되면, 그것을 가장 효과적으로 구현할 수 있는 자료 구조 선택 능력이 중요!! **data structure**에는 변수(나열형), 배열, 리스트, 스택, 큐, 트리, **hash, graph** 등

# 1차원 배열의 선언 (5/10)

## ■ 배열 사용 시 주의할 점

- ✓ 배열의 시작 인덱스는 0.
- ✓ **Boundary check**: 배열의 경계 검사는 프로그래머가 담당.
- ✓ C에서 배열 전체 치환은 지원 안됨

```
#include <stdio.h> // 배열 주의 사항

int main()
{
    int j=3;
    int num[3], i, sum = 0;
    int num2[3];

    num[0] = 1; num[1] = 1; num[2] = 1;

    for (i=0; i<=3; i++)
        sum += num[i];

    printf("%d\n", sum);

    num2 = num; // error. Then, how?
}
```

# 1차원 배열의 선언 (6/10)

- 배열 예제: 한달 동안 매일 정오의 기온을 입력 받아 평균 기온과 가장 추운 날과 더운 날 구하기
  - ✓ Divide And Conquer 접근 방법 사용

```
#include <stdio.h>
int main()
{
    int i, temp[31], min, max, avg, days;
```

```
    printf("How many days in the month? :");
    scanf("%d", &days); // days 값이 양수가 아니면 예외 처리
    for (i=0; i<days; i++) {
        printf("Enter temperature for day %d: ", i+1);
        scanf("%d", &temp[i]);
    }
```

```
    avg = 0;
    for (i=0; i<days; i++)
        avg += temp[i];
    printf("Average temperature : %d\n", avg/days);
```

?

```
    printf("Minimum Temperature : %d\n", min);
    printf("Maximum Temperature : %d\n", max);
```

```
}
```



## ■ 배열 예제

```
#include <stdio.h>
#include <unistd.h>
#include <termios.h>

int main(void)
{
    int i;
    char mess[80];
    struct termios old_tio, new_tio;
    tcgetattr(STDIN_FILENO,&old_tio);
    new_tio=old_tio;
    new_tio.c_lflag &=(~ICANON );
    tcsetattr(STDIN_FILENO,TCSANOW,&new_tio);
    printf("Enter Message (less than 80 characters) \n");
    for (i=0; i<80; i++) {
        mess[i] = getchar();
        if (mess[i] == '\n')
            break;
    }
    printf("\n");
    for (i=0; mess[i] != '\n'; i++)
        printf("%c", mess[i]);
    printf("\n");
    tcsetattr(STDIN_FILENO,TCSANOW,&old_tio);
    return 0;
}
```

## ■ 배열 예제

```
// 숫자 정렬
#include <stdio.h>   #include <stdlib.h>   #include <time.h>

int main()
{
    int org[5], cook[5], tmp;
    int i, j;

    srand(time(NULL));

    for (i=0; i<5; i++)
        cook[i] = org[i] = rand() % 45 + 1;

    for (i = 1; i < 5; i++) {
        for (j = 0; j < 5 - i; j++) {
            if (cook[j] > cook[j + 1]) { // swap code
                tmp = cook[j];
                cook[j] = cook[j + 1];
                cook[j + 1] = tmp;
            }
        }
    }

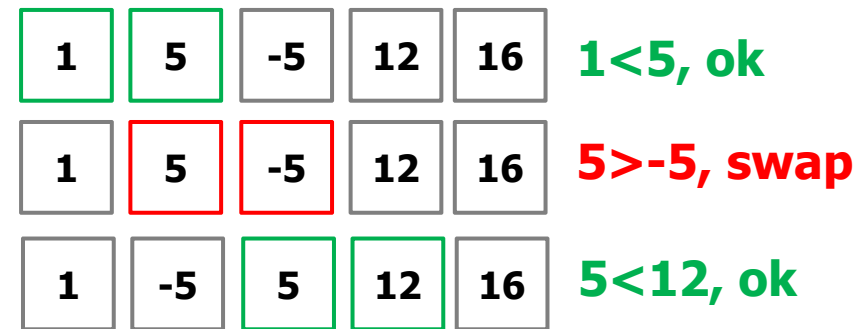
    for (i=0; i<5; i++)
        printf("%d\t%d\n", org[i], cook[i]);
}
```

# 1차원 배열의 선언 (9/10)

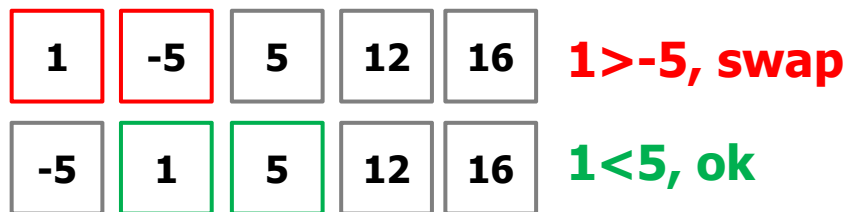
- bubble sorting 상세 설명
  - Cook[5] = {5, 1, 12, -5, 16}



<Step 1>



<Step 2>



<Step 3>



<Step 4>

- 정렬(sorting)은 실제 프로그램에서 많이 사용됨
  - ✓ 성적 처리
  - ✓ 주소록 정렬
  - ✓ 이진 탐색을 위한 데이터 관리
  - ✓ 최단 거리 (shortest path)발견
  - ✓ 등등..
- 정렬의 방법
  - ✓ bubble sorting
  - ✓ insertion sorting
  - ✓ selection sorting
  - ✓ quick sorting
  - ✓ merge sorting
  - ✓ ...

→ Data Structure 수업에서 직접 작성해 보시길...

## ■ 문자열(string)

- ✓ Null Character('\0')로 종료되는 문자 유형(char type)의 1차원 배열
- ✓ “ ”로 묶인 문자들의 집합(컴파일러가 자동으로 '\0' 추가)
- ✓ 문자열을 저장할 배열의 크기는 문자열의 크기 + 1

```
..... (9page 예제의 일부 수정)

printf("Enter Message (less than 80 characters) \n");
for (i=0; i<80; i++) {
    mess[i] = getchar();
    if (mess[i] == '\n')
        break;
}
mess[i] = '\0';

printf("\n");
for (i=0; mess[i] != '\0'; i++)
    printf("%c", mess[i]);
printf("\in");

puts(mess);
printf("%s\n", mess);

..... (9page 예제의 일부 수정)
```

n i c e \n

nice\n가 입력되면..

1차원 문자 배열이 문자열로 바뀜

'\0'가 없다면?

## ■ 문자열 관련 라이브러리

### ✓ 입출력

- gets(str), puts(str): 입출력 처리(‘\r’과 ‘\0’간에 변환)

### ✓ 문자열 관리

- strcpy(to\_str, from\_str) : 문자열 복사
- strcat(to\_str, from\_str) : 문자열 연결
- strcmp(str1, str2) : 문자열 비교 (사전식 순서)
- strlen(str) : 문자열 길이 복귀 (‘\0’ 제외)

strlen(), strcpy(), strcmp(), strcat()  
를 사용하기 위해 포함

Thank you\r를 입력하면?

```
#include <stdio.h> // 문자열 입출력 예제
#include <string.h> // strcpy()를 사용할 때 필요

main()
{
    char str1[80], str2[80];

    printf("Enter a string (less than 80 chars) : \n");
    gets(str1);
    strcpy(str2, str1);
    puts(str2);
}
```

for (i=0; str2[i]; i++)  
 printf("%c", str2[i]);  
or  
printf("%s", str2);

## ■ 문자열 예제

```
#include <string.h>
#include <stdio.h>

main()
{
    char str1[80], str2[80];
    int i;

    printf("Enter the first string: "); gets(str1);
    printf("Enter the second string: "); gets(str2);

    printf("%s is %d chars long\n", str1, strlen(str1));
    printf("%s is %d chars long\n", str2, strlen(str2));

    i = strcmp(str1, str2);
    if (!i) printf("The string are equal\n");
    else if (i < 0) printf("%s is less than %s\n", str1, str2);
    else printf("%s is greater than %s\n", str1, str2);

    if (strlen(str1) + strlen(str2) < 80) {
        strcat(str1, str2); printf("%s\n", str1);
    }
}
```

Thanks God\r'가 입력되고

Its Friday\r'가 입력

## ■ 문자열 예제 (개발자가 흔히 하는 실수)

```
#include <string.h> // 배열 예제.. be cautious
#include <stdio.h>

main()
{
    char limit = 'h';
    char str[8];
    int i, count = 0;

    gets(str);

    for (i=0; i<(int)strlen(str); i++) {
        if (str[i] < limit)
            count++;
    }

    printf("[%d] characters are smaller than %c\n", count, limit);
}
```

“harry”가 입력되면?

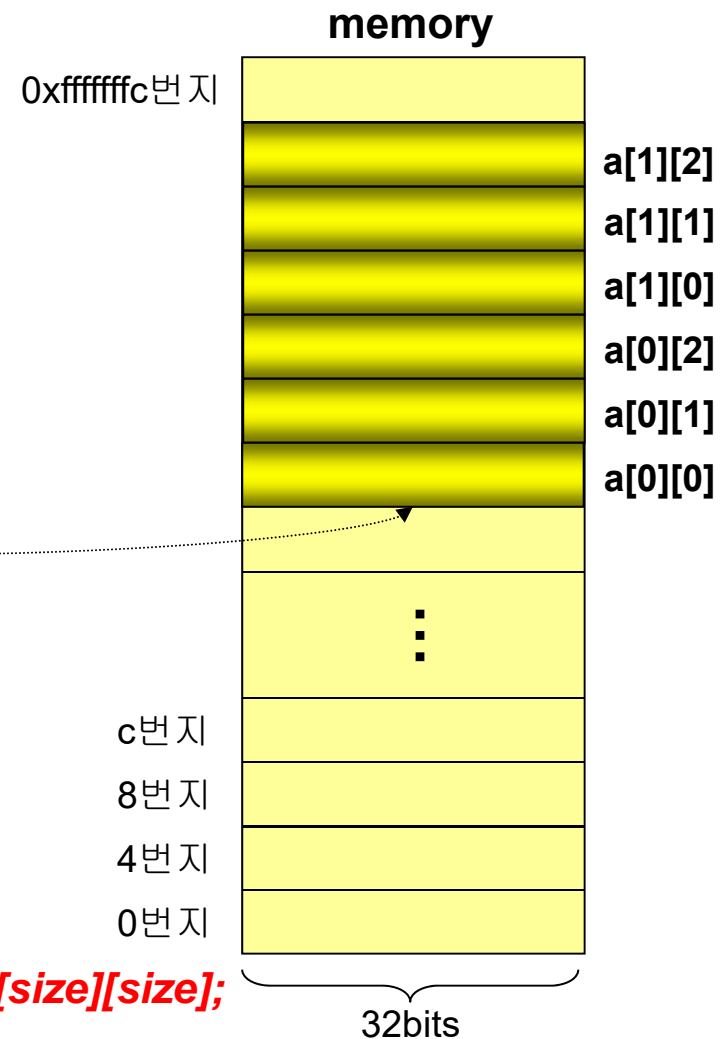
“harrypotter”가 입력되면?



- 이차원 배열
- 형식

```
type variable_name[size][size];
```

```
int a[2][3];
```



☞ **a == &a[0][0], a[1] == &a[1][0].**

☞ 삼차원 배열 **type variable\_name[size][size][size];**

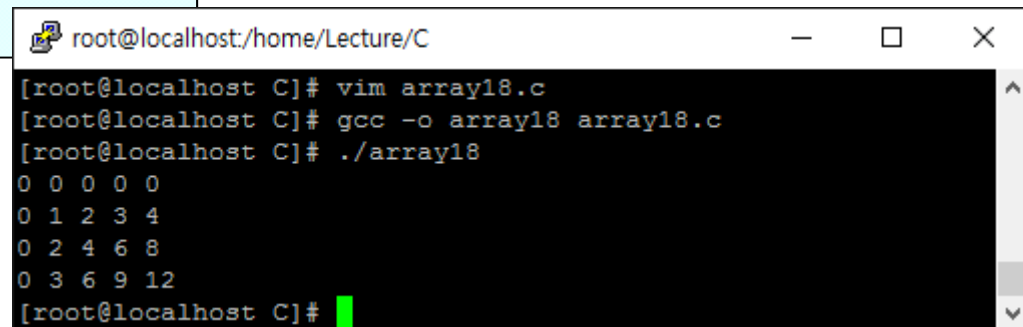
## ■ 이차원 배열의 예

```
#include <stdio.h>

main()
{
    int i, j;
    int twod[4][5]; // 4X5 행렬

    for (i=0; i<4; i++)
        for (j=0; j<5; j++)
            twod[i][j] = i*j;

    for (i=0; i<4; i++) {
        for (j=0; j<5; j++)
            printf("%d ", twod[i][j]);
        printf("\n");
    }
}
```



```
root@localhost:/home/Lecture/C
[root@localhost C]# vim array18.c
[root@localhost C]# gcc -o array18 array18.c
[root@localhost C]# ./array18
0 0 0 0 0
0 1 2 3 4
0 2 4 6 8
0 3 6 9 12
[root@localhost C]#
```

## ■ 함께 해 봅시다:

- ✓ 2과목을 수강하는 6명의 학생의 시험 점수를 관리하기 위한 프로그램의 템플릿

```
#include <stdio.h>
#define MAX_STUDENTS    6
#define NR_OF_SUBJECT  2
int score[MAX_STUDENTS][NR_OF_SUBJECT];
int main(void)
{
    int i, j;
    for (i=0; i<6; i++) {
        for(j=0; j<2; j++) {
            printf("[%d]번째 학생의 [%d]번째 과목 성적은?", i+1, j+1);
            scanf("%d", &score[i][j]);
        }
    }
    printf("====Listing====\n");
    for (i=0; i<6; i++) {
        printf("[%d]번째 학생:", i+1);
        for (j=0; j<2; j++)
            printf("\t [%d] 번째 과목 성적: [%d]", j, score[i][j]);
        printf("\n");
    }
    return 0;
}
```

# 배열의 초기화 (1/3)

## ■ 형식

```
type variable_name[size]={value, value, ..., value};
```

```
#include <stdio.h>

int main(void)
{
    int arr1[5]={1, 2, 3, 4, 5};
    int arr2[]={1, 2, 3, 4, 5, 6, 7};
    int arr3[5]={1, 2};
    int ar1Len, ar2Len, ar3Len, i;

    ar1Len = sizeof(arr1) / sizeof(int); // 배열 arr1의 길이 계산
    ar2Len = sizeof(arr2) / sizeof(int); // 배열 arr2의 길이 계산
    ar3Len = sizeof(arr3) / sizeof(int); // 배열 arr3의 길이 계산

    for(i=0; i<ar1Len; i++)
        printf("%d ", arr1[i]);
    printf("\n");

    for(i=0; i<ar2Len; i++)
        printf("%d ", arr2[i]);
    printf("\n");

    for(i=0; i<ar3Len; i++)
        printf("%d ", arr3[i]);
    printf("\n");
    return 0;
}
```

## ■ 이차원 배열의 초기화

```
#include <stdio.h> // 2차원 배열 초기화 예제
int main()
{
    int i, j;
    int a[2][3] = {1, 2, 3, 4, 5, 6};
    char ch;

    for (i=0; i<2; i++) {
        for (j=0; j<3; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }

    printf("%d\n", sizeof(ch));
    printf("%d\n", sizeof(i));
    printf("%d\n", sizeof(a[0][0]));
    printf("%d\n", sizeof(a));
}
```

int a[2][3] = {{1, 2, 3}, {4, 5, 6}};  
or  
int a[2][3] = {{1, 2}, {4, 5}};

## ■ 배열의 초기화 시 크기 없는 배열(unsized array) 사용 가능

```
#include <stdio.h>
#include <string.h>

int main()
{
    char prompt[] = "Enter command";
    int a[] = {1,4,9,16,25};
    int b[][3] = {1,4,9,16,25,36};

    printf("%s\n", prompt);
    printf("%d\n", a[4]);
    printf("%d %d %d\n", b[0][0], b[0][2], b[1][0]);

    printf("%d, %d, %d\n", sizeof(prompt[0]), sizeof(prompt), strlen(prompt));
    printf("%d, %d\n", sizeof(a), sizeof(a[0]));
    printf("%d, %d, %d\n", sizeof(b), sizeof(b[0][0]), sizeof(b[0]));
}
```

b[2][] 또는 b[][]는 사용 불가

C에서는 맨 왼쪽 차원만  
크기 미 지정 가능

- 문자 유형의 이차원 배열 (문자열의 1차원 배열)
- 예제: 학번을 입력 받아 대응되는 학생 이름을 출력

```
#include <stdio.h>
int main()
{
    char names[10][20];
    int i, query;

    for (i=0; i<10; i++) {
        printf("%d번 이름을 입력하세요 : ", i+1);
        gets(names[i]);
    }

    do {
        printf("학생의 번호 입력 : ");
        scanf("%d", &query);
        query--;
        if ((query < 0) || (query > 9)) {
            printf("학생이 없습니다\n");
            break;
        }
        printf("%d번의 이름은 %s\n", query+1, names[query]);
    } while (1);
}
```

## ■ 예제: 전자 사전

```
#include <string.h>
#include <stdio.h>

char words[][2][40] = {
    "grain", "곡물",
    "grand", "중요한, 화려한",
    "grant", "허락하다",
    "grape", "포도",
    "grasp", "잡다, 쥐다"
    "", ""
};

int main()
{
    char query[12];
    int i;

    printf("Enter English word: ");
    gets(query);

    for (i=0; strcmp(words[i][0], ""); i++)
        if (!strcmp(words[i][0], query)) {
            printf("==> %s\n", words[i][1]);
            break;
        }

    if (!strcmp(words[i][0], ""))
        printf("Not in dictionary\n");
}
```



- 배열을 선언 방법을 이해
- 각 배열 원소를 접근하는 방법을 이해
- 문자열의 특징을 이해
- `gets`, `puts`, `strcpy`, `strcmp`, `strlen`, `strcat` 사용 방법을 이해
- 다차원 배열을 이해
- 배열의 초기화 방법을 이해
- 문자열을 배열 원소로 사용하는 방법을 이해

## ■ 주소록 작성

- ✓ 이름과 Cellular Phone 전화 번호로 구성된 주소록 작성
- ✓ 5명의 이름과 전화 번호는 미리 초기화
- ✓ 최대 20의 이름과 번호 추가 기능
- ✓ 이름과 전화 번호 list 기능
- ✓ 이름으로 검색 기능
- ✓ 이름으로 정렬 기능
- ✓ help 기능
  
- ✓ Bonus: 이름 삭제 기능, 전화 번호로 정렬, 전화 번호로 검색

```
#include <stdio.h>
#include <string.h>

// 자료 구조 초기화
#define MAX_ENTRY 25
#define MAX_CHARS 20

char address[MAX_ENTRY][2][MAX_CHARS] = {
    "유비", "011-111-1111",
    "관우", "017-111-1111",
    "장비", "016-111-1111",
    "조자룡", "011-777-7777",
    "제갈량", "019-111-1111",
};

int current_index = 5;

// 내부 함수
int addr_add(); // 정상적인 경우 0 복귀, 문제 발생의 경우 -1 복귀
int addr_help();
int addr_sort();
int addr_list();
int addr_search();
```

```
// 메인 함수
main()
{
    char cmd[20];

    printf("My address program\n");
    while (1) {
        printf("Enter any command (help to show commands) : ");
        gets(cmd);

        if (!strcmp(cmd, "quit")) {
            printf("Have a nice day..\n");
            break;
        }
        else if (!strcmp(cmd, "help"))
            addr_help();
        else if (!strcmp(cmd, "sort"))
            addr_sort();
        else if (!strcmp(cmd, "list"))
            addr_list();
        else if (!strcmp(cmd, "add"))
            addr_add();
        else if (!strcmp(cmd, "search"))
            addr_search();
        else {
            printf("잘못된 명령어 입니다\n");
            addr_help();
        }
    }
}
```