

# 컴퓨터 프로그래밍2

## 7주차\_실습

---

충남대학교 컴퓨터공학과  
장서윤

[pineai@cnu.ac.kr](mailto:pineai@cnu.ac.kr)

# 다차원 배열

- ▶ 1차원 배열처럼 메모리의 연속된 저장공간이 할당

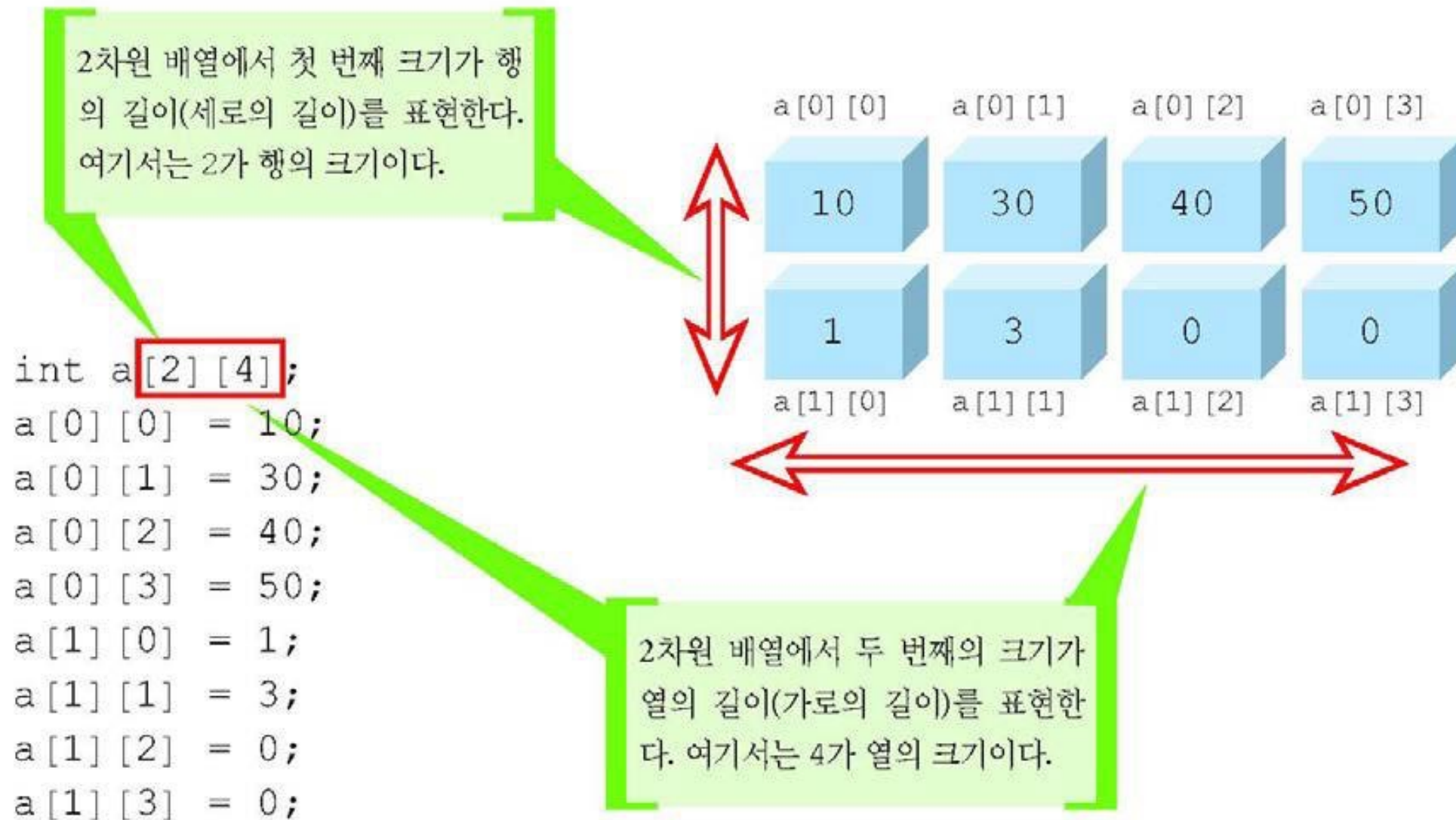


그림 12.6 2차원 배열의 개념

# 다차원 배열

---

## ▶ 2차원 배열의 선언

- ▶ 초기화 없이 선언만 한다면 배열의 크기는 행과 열, 두 개 모두를 반드시 지정해야 함
- ▶ 초기값을 저장한다면 배열의 크기는 지정하지 않아도 됨

```
int score1[4];  
int score2[4];  
int score3[4];
```

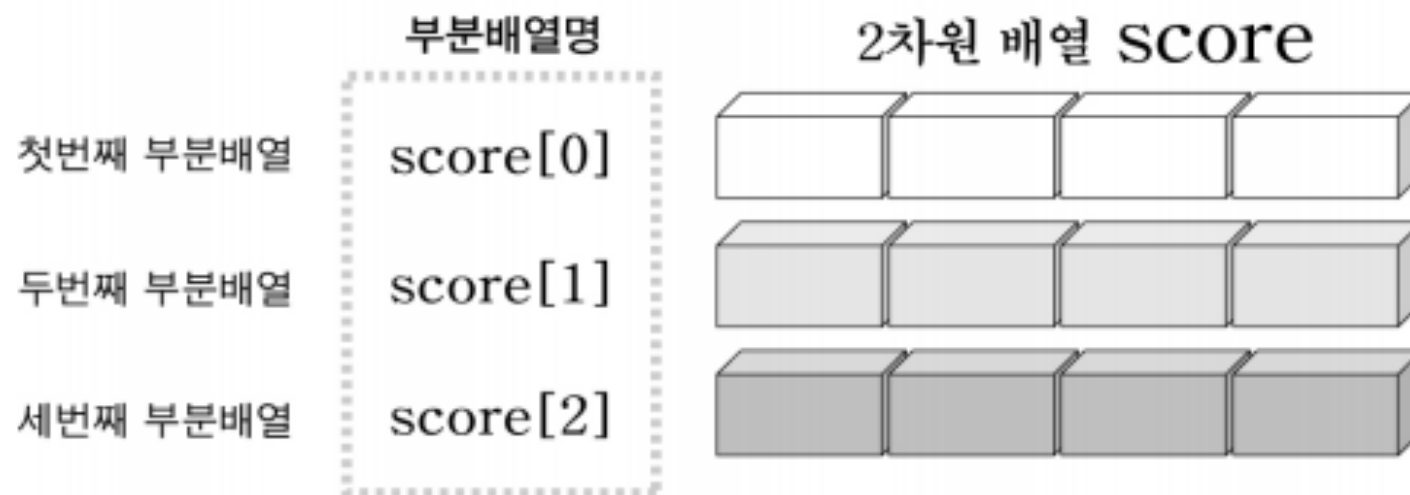
} 같은 형태의 1차원 배열이 3개 필요하다.



# 다차원 배열

---

- ▶ 2차원 배열의 선언
  - ▶ 1차원 배열과 동일하게 배열명과 첨자를 사용하여 선언
  - ▶ 전체 배열은 12개의 int형 기억 공간이 1차원 배열의 형태로 할당
  - ▶ 2차원 배열은 논리적으로 행렬로 표현



```
int score[3][4];
```

## 배열 요소의 형태

int형 변수 4개짜리 1차원 배열이 2차원 배열의 요소가 됨

# 다차원 배열

---

## ➤ 2차원 배열의 선언

➤ `int a[2][4] = {10, 30, 40, 50, 1, 3, 0};`

➤ `int b[][4] = {10, 30, 40, 50, 1, 3, 0};`

➤ `int c[2][4] = {{10, 30, 40, 50}, {1, 3, 0, 0}};`

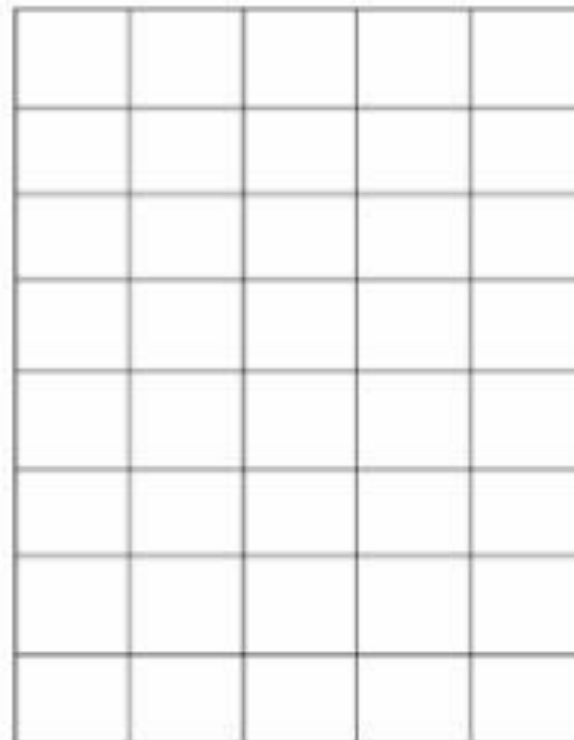
# 다차원 배열

## ▶ 3차원 배열의 선언

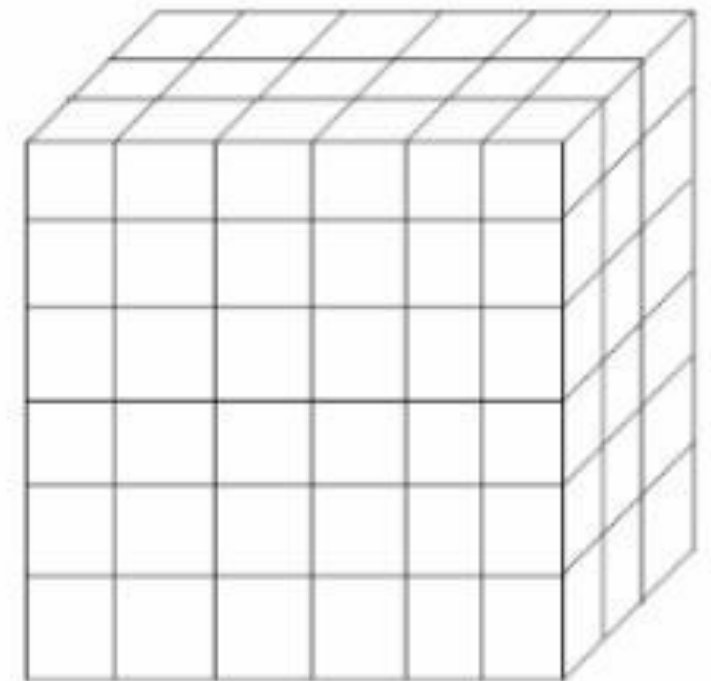
```
int a[2][3][4] = {  
    {  
        {10, 30, 40, 50},  
        {1, 3, 0, 0},  
        {2, 3, 4, 5}  
    },  
    {  
        {10, 30, 40, 50},  
        {1, 3, 0, 0},  
        {2, 3, 4, 5}  
    }  
}
```



1차원 배열



2차원 배열



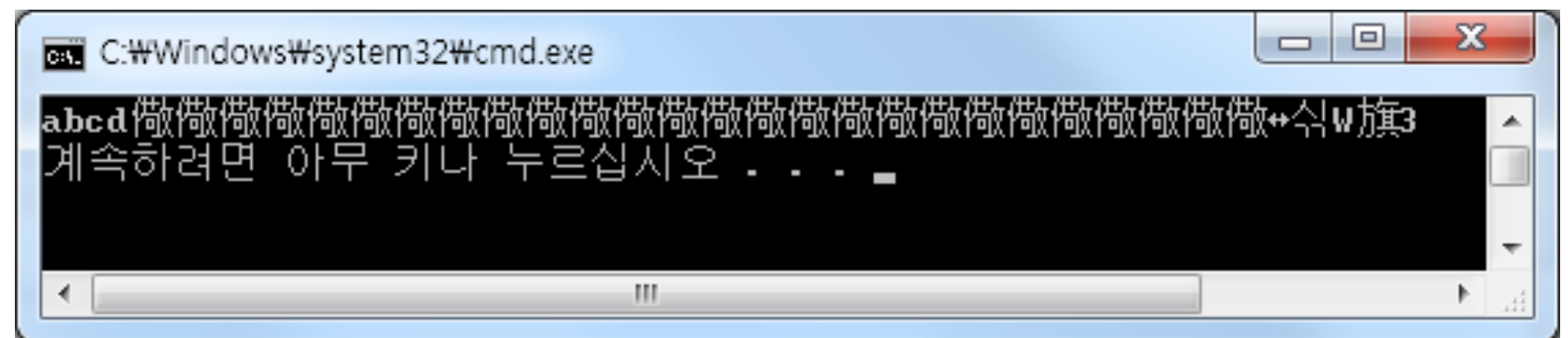
3차원 배열

# 문자열 배열

---

- ▶ 널 (NULL) 문자('\0')가 문자 배열의 마지막에 저장
- ▶ 문자 배열의 마지막에 널 문자가 없으면 저장된 문자열의 끝을 알 수 없음

```
int main(void) {  
  
    char word[50];  
  
    word[0] = 'a';  
    word[1] = 'b';  
    word[2] = 'c';  
    word[3] = 'd';  
    printf("%s\n", word);  
  
    return 0;  
}
```



# 실습 1

---

## ➤ 2차원 배열에 데이터를 입력받고 출력하기

### ➤ 반복문을 이용하여 출력

**출력결과**

1 2 3  
4 5 6  
7 8 9

```
int array[3][3];
int rowIndex, colIndex;

for (rowIndex = 0; rowIndex < 3; rowIndex++)
    for(colIndex = 0; colIndex < 3; colIndex++)
        scanf("%d",&array[rowIndex][colIndex]);

for (rowIndex = 0; rowIndex < 3; rowIndex++){
    for(colIndex = 0; colIndex < 3; colIndex++)
        printf("%d ", array[rowIndex][colIndex]);
    printf("\n");
}
```



## 실습 2

---

- ▶ 학생의 숫자를 입력받고 학생 숫자 만큼 국어, 영어, 수학 성적을 2차원 배열에 저장하여 출력하는 프로그램
- ▶ 학생 숫자는 20까지 입력 가능

```
입력 받을 학생수를 입력하세요 : 3
국어 성적 입력 : 1
영어 성적 입력 : 2
수학 성적 입력 : 3
국어 성적 입력 : 4
영어 성적 입력 : 5
수학 성적 입력 : 6
국어 성적 입력 : 7
영어 성적 입력 : 8
수학 성적 입력 : 9
[0] 1,2,3
[1] 4,5,6
[2] 7,8,9
```

입출력 예시

# 실습 2

---

## ➤ Main 함수

```
int scoreArray[20][3];
int inputStudentSize;
int studentIndex;

printf("입력 받을 학생수를 입력하세요 : ");
scanf("%d",&inputStudentSize);

for(studentIndex = 0; studentIndex < inputStudentSize;
    studentIndex++)
    inputScore(&scoreArray[studentIndex],inputStudentSize);

for(studentIndex = 0; studentIndex < inputStudentSize;
    studentIndex++)
    printScore(&scoreArray[studentIndex], studentIndex,
        inputStudentSize);

return 0;
```

권장 형태가 아니므로 semantic Issue가 발생함

## 실습 2

---

- void inputScore(int \*scoreArray, int inputIndex);
- void printScore(int \*scoreArray, int index, int inputIndex);

```
void inputScore(int *scoreArray, int inputIndex){
    printf("국어 성적 입력 : ");
    scanf("%d",&scoreArray[0]);
    printf("영어 성적 입력 : ");
    scanf("%d",&scoreArray[1]);
    printf("수학 성적 입력 : ");
    scanf("%d",&scoreArray[2]);
}

void printScore(int *scoreArray, int index, int inputIndex){
    printf("[%d] %d,%d,
           %d\n",index,scoreArray[0],scoreArray[1],scoreArray[2]);
}
```

# 과제 1

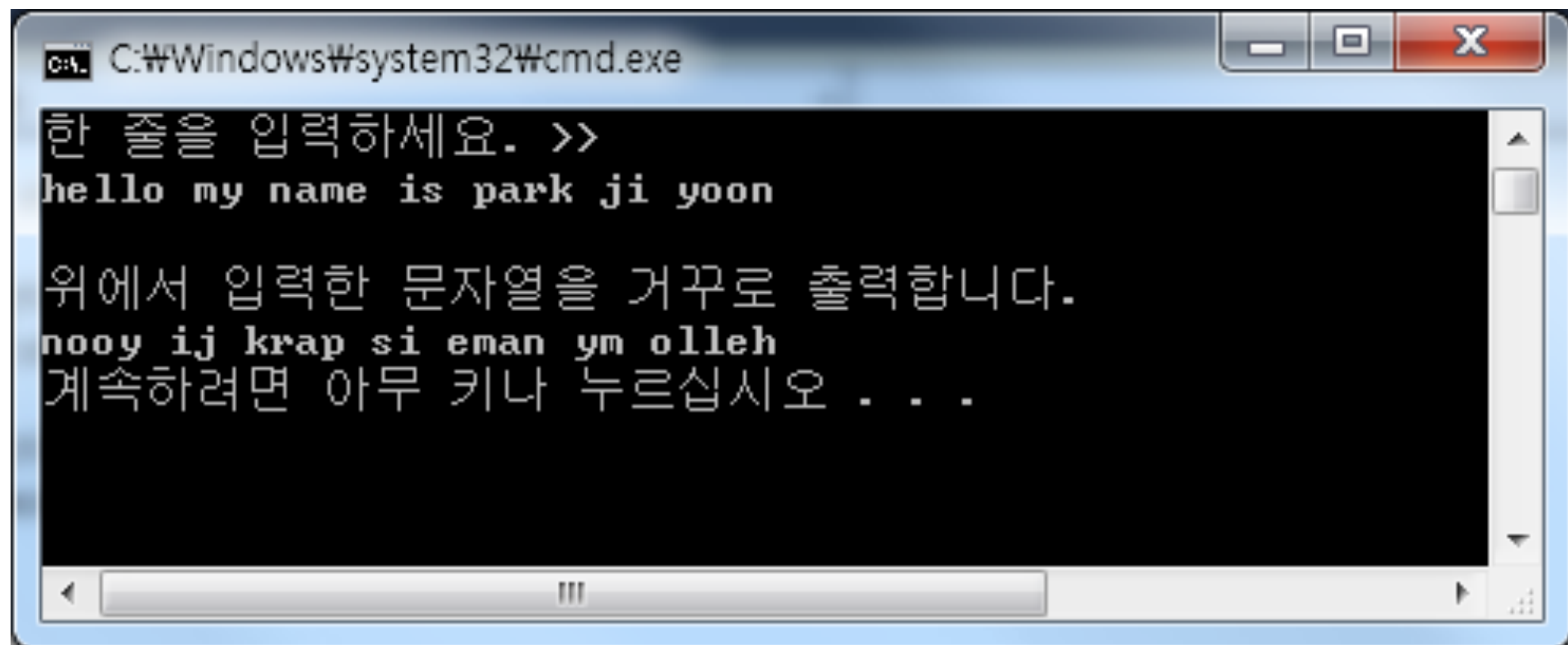
---

- ▶ 실습 2에 각각 학생의 총점과 평균을 구해서 출력하는 메소드를 구현하고, 메소드 호출을 위해 main 함수를 수정하시오.
- ▶ `int printSum(int *scoreArray, int inputIndex)`
  - ▶ 학생의 총점 (영어, 수학, 국어)을 구해서 출력하고 반환하는 함수
- ▶ `int printAvg(int *scoreArray, int inputIndex)`
  - ▶ 학생의 평균을 구해서 출력하고 반환하는 함수

## 과제 2

---

- ▶ 키보드로부터 입력받은 한 줄의 문자열을 역순으로 출력하는 프로그램 작성
- ▶ `void invertedOutput(char str[])`
  - ▶ 역순으로 만들어서 출력하는 메소드



```
C:\Windows\system32\cmd.exe

한 줄을 입력하세요. >>
hello my name is park ji yoon

위에서 입력한 문자열을 거꾸로 출력합니다.
nooy ij krap si eman ym olleh
계속하려면 아무 키나 누르십시오 . . .
```

## 과제 3

---

- ▶ 배열에 저장된 데이터를 선택정렬을 이용하여 오름차순으로 정렬하는 프로그램을 작성
  - ▶ `int* makeArray(int *intArray, int inputSize)`
    - ▶ 지난 주 과제에서 사용한 메소드를 그대로 사용
  - ▶ `void swap(int * x, int * y)`
    - ▶ 받은 두 값의 위치를 변경하는 메소드
  - ▶ `void sortArray(int *intArray, int inputSize)`
    - ▶ 받은 배열을 선택 정렬하는 메소드

# SELECTION SORT

---

- ▶ 반복적으로 특정 값을 정렬된 최종 위치에 배치시킴으로써 값들의 목록을 정렬

3 9 6 1 2



3에서 시작하여 오른쪽 방향으로 조사한다.  
1이 가장 작은 값이다. 1과 3을 교환한다.

1 9 6 3 2



9에서 시작하여 오른쪽 방향으로 조사한다.  
2가 가장 작은 값이다. 9와 2를 교환한다.

1 2 6 3 9



6부터 시작하여 오른쪽으로 조사한다.  
3이 가장 작은 값이다. 6과 3을 교환한다.

1 2 3 6 9



6부터 시작하여 오른쪽으로 조사한다.  
6이 가장 작은 값이다. 6과 6을 교환한다.

1 2 3 6 9



# SELECTION SORT

---

## ➤ 알고리즘

- ① 목록 전체를 조사하여 가장 작은 값을 찾는다.
- ② 이 값과 목록의 첫 번째 위치에 있는 값을 교환한다.
- ③ 목록의 나머지 값들(첫 번째 값을 제외한 모든 값)을 조사하여 가장 작은 값을 찾고, 이 값과 목록의 두 번째 위치에 있는 값을 교환한다.
- ④ 목록의 나머지 값들(처음 두 번째 값을 제외한 모든 값)을 조사하여 가장 작은 값을 찾고, 이 값과 목록의 세 번째 위치에 있는 값을 교환한다.
- ⑤ 목록의 각 위치에 대해서 이러한 과정을 계속한다.



# 추가 과제

---

- ▶ 미로에서 최단 경로 찾기
  - ▶ 이진 직사각형 행렬의 형태로 미로
  - ▶ 경로는 값 1을 갖는 셀로만 구성 될 수 있음
  - ▶ 주어진 순간에 4방향중 하나에서 한단계로만 이동
  - ▶ 올바른 이동은 다음과 같음
    - ▶ 위로 이동  $(x,y) \rightarrow (x-1, y)$
    - ▶ 오른쪽으로 이동  $(x,y) \rightarrow (x, y+1)$

Source = (0,0)  
Destination = (7,5)  
최단 경로 12

1	1	1	1	1	1	0	0	1	1	1
0	1	1	1	1	1	1	0	1	0	1
0	0	1	0	1	1	1	1	0	0	1
1	0	1	1	1	0	1	1	1	0	1
0	0	0	1	0	0	0	0	1	0	1
1	0	1	1	1	0	0	1	1	0	
0	0	0	0	1	0	0	1	0	1	
0	1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	0	0	1	1	1	1
0	0	1	0	0	1	1	0	0	1	

# 과제제출

---

- ▶ 사이버캠퍼스에 제출
- ▶ 파일 이름 : [CP2]07\_본인학번
- ▶ 제출 기한 : 5월 3일 23:59까지
  
- ▶ 소스에는 주석 작성이 되어 있어야 함
- ▶ 실습은 수행 결과를 [스크린샷]하여 결과에 대한 설명을 적어 제출 (제출 방법은 상관 없음)
- ▶ 과제는 소스와 스크린샷을 제출
- ▶ 실습과 과제 모두 압축하여 하나의 파일로 제출
- ▶ 추가과제는 메일로 직접 보낼 것 (메일 제목은 파일 이름과 동일)