프논: 함수와 배열 복습

■ Review : 함수

	void 함수	Value returning 함수
호출	① 단독으로 쓰인다 ② 인수를 매개변수와 일치	⑤ (단독 혹은) **문장 안에서** ⑥ 인수를 매개변수와 일치
정의	③ 매개변수를 선언 ④ return; // 값없음!!	⑦ 매개변수를 선언 **⑧ return 값;**

```
void printProduct(int a, int b);

#include <stdio.h>
int main(void)
{
    printProduct(10, 20); // 호출
}

void printProduct(int a, int b)
{
    printf("%d\n", a * b);
    return;
}
```

```
int calProduct(int a, int b);

#include <stdio.h>
int main(void)
{
    printf("%d\n", calProduct(10, 20);
}

int calProduct(int a, int b)
{
    return a * b;
}
```

■ Review : 배열

배열을 입력, 출력, 처리할 때 **항상 원소단위**로 해야 함을 잊지 말자.
즉 **for문을 사용해야** 하는 경우가 대부분

SIZE를 배열의 크기라 가정할 때
int A[SIZE], B[SIZE];

	절대 해서는 안 되는 코드	방법
배열을 입력할 때	**scanf("%d", %A[SIZE]);**	for (i = 0; i < SIZE; i++) scanf("%d", &A[i])
배열을 처리 예: 배열A를 배열B로 복사	**B[SIZE] = A[SIZE];**	for (i = 0; i < SIZE; i++) B[i] = A[i];
배열을 출력할 때	**printf("%d", A[SIZE]);**	for (i = 0; i < SIZE; i++) printff("%d", A[i])

프논 12th 수업에서 다루어지는 것들- **배열을 함수 매개변수로 전달하기.**

크기가 5인 정수형 배열의 원소들을 출력하려한다. main함수만으로 작성하면

```
int main(void)
{
    int arr[5] = {10, 20, 30, 40, 50};
    int i;

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

```
int main(void)
{
    int x;

    함수1(x);

}

void 함수1(int a)
{
...
}
```

이제 위의 프로그램을 **배열을 함수 매개변수로 하는** printArray함수를 정의, 호출하여 작성해보자.

```
void printArray(int a[]);
int main(void)
{
    int arr[5] = {10, 20, 30, 40, 50};

    printArray(arr); // 호출시 인수로는 배열이름을 사용한다.
}
```

정의부분의 매개변수인 a 는
사실상 포인터(pointer)이다.
뒤에서(2학기에!!) 배운다.

지금은 그냥 배열의 이름처럼
사용한다고 생각하자

그런데,
위의 프로그램은 좋은 프로그램은 아니다. 배열을 함수 매개변수로 전달하는 경우, 대부분 그 **크기**를 같이 전달하는 것이 좋다. 즉 printArray 함수를 아래와 같이 **수정**할 수있다. 앞으로는 이런 형식으로 사용하도록 한다.

```
// 더 좋은 코드
void printArray(int a[], int size);

int main(void)
{
    int arr[5] = {10, 20, 30, 40, 50};
    printArray(arr, 5); // 호출할 때의 인수에서는 배열의 이름과 배열의 크기
}

void printArray(int a[], int size)  // 정의시 매개변수에는 배열이름[], 배열의크기를 받는 size변수
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", a[i]);
    printf("\n");
    return;
}
```

배열의 함수 매개변수 전달
✓ 호출: 인수는 배열이름, 배열크기(예: printArray(arr, 5)
이때 배열크기는 5 혹은 5로 정의한 심볼릭 상수(예: SIZE)를 써도 되고

✓ 정의: 매개변수는 배열형식의 선언, 크기(예: void printArray(in a[], int size) {…}

■ **연습 a** 위의 printArray 함수(더 좋은 코드)를 사용하고 아래의 출력이 나오게 프로그램을 완성하라.

```
#include <stdio.h>
void printArray(int a[], int size);
int main(void)
{
    int list1[5] = {10, 20, 30, 40, 50};
    int list2[3] = {100, 200, 300};

    // 위의 printArray함수를 사용하여 위의 두 배열의 요소를 출력하라


}
void printArray(int a[], int size) // 위의 코드 그대로
{  ... }
```

```
C:₩WINDOWS₩system32₩cmd.exe
10 20 30 40 50
100 200 300
계속하려면 아무 키나 누르십시오 . . .
```

■ **연습b**

```
#include <stdio.h>
void printArray(int a[], int size);
void changeArray(int b[], int size);
int sumArray(int c[], int size);
int main(void)
{
    int data[5] = {10, 20, 30, 40, 50};
    printArray(data, 5); // 호출할 때의 인수에서는 배열의 이름을 사용한다.

    changeArray(data, 5); // 5대신에 sizeof(data) / sizeof(int)를 써도 된다.
    printArray(data, 5);
    printf("배열의 합은 %d\n", sumArray(data, 5));
}
void printArray(int a[], int size) // 앞에 것 그대로
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", a[i]);
    printf("\n");
    return;
}
void changeArray(int b[], int size) // 배열 b의 각 원소 값을 10배로 바꾼다
{




}
int sumArray(int c[], int size)
{




}
```

```
C:₩windows₩system32₩cmd.exe
10 20 30 40 50
100 200 300 400 500
배열의 합은 1500
계속하려면 아무 키나 누르십시오 . . .
```

■ **생각해보는 예제**

```
#include <stdio.h>
void printArray(int a[], int size);
void test(int arr[], int size, int num);
int main(void)
{
    int data[5] = {10, 20, 30, 40, 50};
    int number = 10;

    printf("배열 data는");
    printArray(data, 5);
    printf("number는 %d\n", number);

    test(data, 5, number);

    printf("배열 data는");
    printArray(data, 5);
    printf("number는 %d\n", number);
}
void printArray(int a[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", a[i]);
    printf("\n");
    return;
}
void test(int a[], int size, int num)   // 매개변수 a를 변화시키는 것은 인수 data를 변화시키는 것
                                        // 그러나 매개변수 num을 변화시켜도 인수 number에는 영향 없음
{
    int i;
    for (i = 0; i < size; i++)
        a[i] *= 10;  // a의 원소들을 10배

    num *= 10; // num을 10배
    return;
}
```

논리야 놀자 #4 – 배열(array)을 이용한 논리 연습

✪ 논리 14: 두 수의 값을 바꾼다.

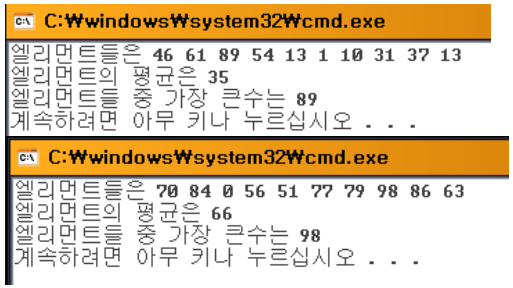
✪ 논리 15: 배열을 역순배열로 바꾼다

LABHW14(배열의 함수 매개변수 전달)

■ **LAB14_1 배열의 함수 매개변수 전달(난이도 중하)**

아래의 실행결과가 나오도록 프로그램을 완성하라.
main함수와 나머지 함수의 정의부분을 완성하면 된다.
난수발생을 위한 seed로 현재의 시간을 이용하는 time()함수를 사용했으므로 실행결과는 매번 다른 수들을 보여준다.

1)2)를 한후, 3), 4)의 순으로 **단계적으로** 완성하도록 하자.



1)0부터 99까지의 난수를 발생시킨다.
2)출력한다
3)평균을 구해서 출력한다
4)가장 큰 수를 구해서 출력하다.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void generateData(int [], int);
void printData(int [], int);
int averageData(int [], int);
int maxData(int [], int);  // 여기에 보이는 원형들처럼 매개변수의 이름을 생략할 수도 있다!

int main(void)
{
    int data[10];  // 계산 수행에 사용할 정적 변수
    int average;
    srand(time(NULL));  // random 값 출력에 사용하는 함수. Seed 값을 부여

    //1)
    generateData(data, 10);
    //2)
    printf("엘리먼트들은 ");
    printData(data, 10);

    //3)
    average = averageData(data, 10);
    printf("엘리먼트의 평균은 %d\n", average);

    //4) 호출부분 추가

}

void generateData(int a0[], int size)
{
    ...
}

void printData(int a2[], int size)
{
    int i;
    for (i = 0; i < size; i++) // Index 0..9까지
        printf("%d ", a2[i]);
    printf("\n");
    return;
}

int averageData(int a1[], int size)
{
   ...
}

int maxData(int a3[], int size)
{
    int i, best = a3[0];
    for (i = 0; i < size; i++) // Index 0..9까지
        if (best < a3[i])
            best = a3[i];
    return best;
}
```

■ **LAB13_1F**

LAB13_1을 사용자정의 함수를 사용하여 다시 작성하라. 함수의 원형 및 설명은 다음과 같다.

```
void printArray(int a[], int size); // 배열 a의 원소들을 출력하는 함수
int equalArray(int a[], int b[], int size); // 배열 a와 b가 같으면 1을 다르면 0을 반환하는 함수
                                            (이때 배열 a와 b는 크기가 같다고 가정한다)
```

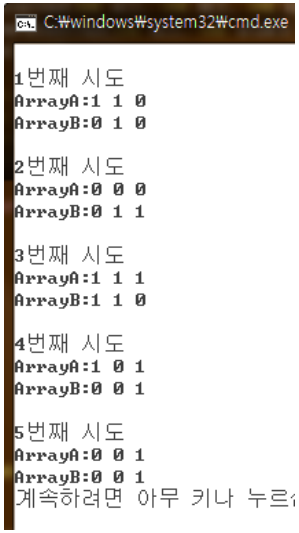
```
// 뼈대 코드
#define SIZE 5
#include <stdio.h>

void printArray(int a[], int size);
int equalArray(int a[], int b[], int size);

int main(void)
{
    int arrayA[SIZE] = {1, 2, 3, 4, 1};
    int arrayB[SIZE];
    int arrayC[SIZE];
    int i;

    for (i = 0; i < SIZE; i++)
    {
        arrayB[i] = arrayA[i];
        arrayC[5 - i - 1] = arrayB[i];
    }

     // 배열 3개를 출력하는 부분 추가: printArray 호출



    if (equalArray(arrayA, arrayB, SIZE)) // if (equalArray(arrayA, arrayB, SIZE) == 1)
        printf("ArrayA와 arrayB는 같다\n");
    else
        printf("ArrayA와 arrayB는 다르다\n");

    if (equalArray(arrayA, arrayC, SIZE))
        printf("ArrayA와 arrayC는 같다\n");
    else
        printf("ArrayA와 arrayC는 다르다\n");
}

void printArray(int a[], int size) // 그대로 사용한다
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", a[i]);
    printf("\n");
    return;
}

// 배열 a와 b가 같으면 1을 다르면 0을 반환하는 함수를 정의한다
// flag 변수 없이 작성해보자!
int equalArray(int a[], int b[], int size)
{
    ...
}
```

■ LAB13_1F 응용

0혹은 1을 갖는 2개의 배열(크기 3)을 생성시켜 출력하고 같은 배열이 생성될 때까지 반복하는 프로그램을 작성하려 한다. 실행결과는 아래와 같다.

힌트:
do while문을 쓸 수 있다. 위의 프로그램처럼 배열이 (->같은가를 체크한다.
same이 1이 될 때까지(즉 두 개의 배열이 같아질 때까지) 반복한다.

srand를 사용하지 않았을 때의 예시는 아래의 오른쪽 실행 예와 같다.
srand를 사용하면 다양한 결과가 나올 것이다. (오른 쪽 밑의 예)



이전 LAB13_1F에서 새로 정의한 2개의 함수는 그대로 사용하고
하나의 함수를 더 추가해서 정의한다. 함수의 원형 및 설명은 다음과 같다.

```
void generateArray(int a[], int size); // 새로이 추가하는 함수
                                       // 배열 a에 0 혹은 1의 값을 넣는다.
void printArray(int a[], int size);
int equalArray(int a[], int b[], int size);
```

```
#define SIZE 3
#include <stdlib.h>
#include <time.h>
#include <stdio.h>

void printArray(int a[], int size);
int equalArray(int a[], int b[], int size);
void generateArray(int a[], int size);

int main(void)
{
    int arrayA[SIZE], arrayB[SIZE];
    int count = 0;

    // srand(time(NULL)); 주어진 실행결과를 확인한 후 주석을 취소하여 여러 번 실행시켜본다
    do
    {
        printf("\n%d번째 시도\n", ++count);
        generateArray(arrayA, SIZE);
        generateArray(arrayB, SIZE);

         // 두 개의 배열 출력 부분 추가
         ...


    } while (!equalArray(arrayA, arrayB, SIZE));
}

void generateArray(int a[], int size)
{
    ...
}

void printArray(int a[], int size) // 앞 문제의 것을 그대로 사용
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", a[i]);
    printf("\n");
    return;
}

int equalArray(int a[], int b[], int size) // 앞 문제의 것을 그대로 사용
{
   ...
}
```

■ LAB13_2F

LAB13_2를 사용자 정의 함수를 사용하여 다시 작성한다
함수 indexSearch를 사용한다. 원형 및 설명은 다음과 같다.

```
int indexSearch(int a[], int size, int searchKey);
// 배열 a안에  searchKey값이있는가를 탐색하여 해당 첨자를 반환한다. searchKey값이 없는 경우는 -1을 반환한다.
```

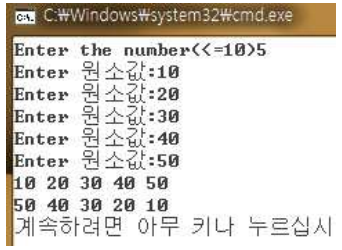
```
#include <stdio.h>
int indexSearch(int a[], int size, int searchKey);
#define SIZE 12

int main(void)
{
    int incomes[SIZE] = {11, 22, 33, 44, 55, 66, 11, 22, 33, 44, 55, 66};
    int income;
    int id;

    printf("탐색할 수입은? ");
    scanf("%d", &income);

    id = indexSearch(incomes, SIZE, income);
    if (id == -1)
        printf("수입 %d를 갖는 달은 없습니다\n", income);
    else
        printf("수입 %d를 갖는 첫번째 달은 %d월입니다\n", income, id + 1);
    return 0;
}

// flag 변수 없이 작성해보자!!
int indexSearch(int a[], int size, int searchKey)
{
    ...
}
```

■ LAB13_3 + LAB13_4F

- □ 단계1: LAB13_4에서 정렬하는 부분을 selectionSort 함수로 정의하여 다시 프로그램 하라.
- □ 단계2: LAB13_3를 수정하여 홀수와 짝수를 정렬하여 출력하라.
  이를 위해 앞의 단계에서 정의한 selectionSort를 LAB13_3의 A), B)에서 호출한다.

■ LAB14_2(역순 배열 만들기)

아래와 같은 실행결과를 갖도록 프로그램을 완성하라.




```
#include <stdio.h>

void printArray(int a[], int size);
void reverse(int a[], int size);
int main(void)
{
    int num;
    int list[10];
    int i;

    printf("Enter the number(<=10)");
    scanf("%d", &num);

    for (i = 0; i < num; i++)
    {
        printf("Enter 원소값:");
        scanf("%d", &list[i]);
    }

    printArray(list, num);
    reverse(list, num);
    printArray(list, num);
}

void printArray(int a[], int size) // 그대로 사용한다
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", a[i]);
    printf("\n");
    return;
}
void reverse(int a[], int size)
{
    …
}
```

■ LAB13_5F(역순 배열 이용하여 이진수 구하기)

LAB13_5 를 사용자 정의함수를 사용하여 다시 작성하려한다

```
#include <stdio.h>

void printArray(int a[], int size);
void reverse(int a[], int size);

int main(void)
{
     int binary[8];
     int num;
     int bIndex = 0; // binary 배열의 인덱스

     printf("Enter 양수(<256):");
     scanf("%d", &num);

     // binary에 이진수를 넣는 코드를 추가, 8의 경우 이진수의 값이 거꾸로 저장. binary[] <- 0001
     // 힌트: while문 사용
     ...

     // binary를 역순으로 바꾼 후, 출력한다(앞서 정의한 reverse, printArray를 호출).
     ...

     return 0;
}

void printArray(int a[], int size)
{
     int i;
     for (i = 0; i < size; i++)
         printf("%d", a[i]); // 공백문자없게 출력하도록 원래의 printArray를 수정
     printf("\n");
     return;
}

void reverse(int a[], int size) // 앞 문제의 정의를 그대로 사용
{
     ...
}
```

LAB 14B - 문자열(추가)

■ **LAB14B_3 문자열의 함수 매개변수 전달**을 살펴보자.

```
void printUpperCase(char s[])
{
	int i;
	for (i = 0; s[i] != '\0'; i++)
		if (s[i] >= 'A' && s[i] <= 'Z')
			printf("%c", s[i]);
	printf("\n");
}
```

이때 일반 배열과는 달리 문자열의 경우는 그 **크기**를 같이 전달 할 필요없다.
그 끝이 '\0'으로 끝나는 성질을 이용하면 되기 때문이다.
아래의 실행결과를 갖도록 함수 strLength를 추가하여 프로그램을 완성하라.

```
C:₩WINDOWS₩system32₩cmd.exe
Enter a string:VeryGood
길이는 8
대문자만 출력하면
VG
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

void printUpperCase(char s[]);
int main(void)
{
	char str[81];

	printf("Enter a string:");
	scanf("%s", str);

	printf("길이는 %d\n", strLength(str)); // strLength 호출

	printf("대문자만 출력하면\n");
	printUpperCase(str); // printUpperCase 호출
}
```

HW 14B- 문자열(추가)

■ **HW14B_4**

인생은 ATTITUDE에 달렸다! - 진대제 전 정보통신부 장관

그가 말하는 `알파벳으로 보는100점짜리 인생의 조건`은 이렇다.
`A 는1, B는2, C는3.....Z는26`같은 방식으로A에서Z까지 점수를 매긴다.
이 공식을 각 단어에 적용하면 된다.
이를테면 행운을 뜻하는 단어
`LUCK`의 경우L(12) +U(21) +C(3) +K(11)`이므로 합계는 47점이 된다.
진 장관은 이렇게 계산한 각 단어의 점수를 프리젠테이션 화면으로
보여주면서 LOVE(사랑)는 낙제점을 면한54점,
돈(MONEY)은 72점
지식(KNOWLEDGE)은96점 밖에 안 된다고 설명한다.
또 열심히 일하는 경우(HARDWORK)도 98점,운도 좋고 돈도 많다는 의미의
`FORTUNE`역시 99점에 불과 하다고 강조한다.
그러나 `자세 몸가짐`을 의미하는 `***ATTITUDE***`는 100점으로
`결국 인생은마음 먹기에 달려있는 것 같다`는게 진대제 전 장관의 강의 요지다.

*휴식,스트레스도 100점
그는 공교롭게도 정신적 중압감을 나타내는`스트레스(***STRESS***)와 휴식을
의미하는 숙어(***TAKEAREST***)역시 100점` 이라며 `인생에는 적당한
스트레스와 휴식도 필요한것 같다`고 덧 붙인다

위의 글에서 설명된대로 단어를 입력받아 그 점수를 계산하는 프로그램을 작성하라.
대문자 소문자 모두 작동하도록 하라.

힌트:
처음에는 대문자에만 작동하도록 만든후,
이를 성공하면 소문자도 다룰수 있도록 수정하여 완성한다.

```
C:₩WINDOWS₩system32₩cmd.exe
단어를 입력하세요<빈칸없이>:Love
점수는 54
계속하려면 아무 키나 누르십시오 . . .
```

```
C:₩WINDOWS₩system32₩cmd.exe
단어를 입력하세요<빈칸없이>:Attitude
점수는 100
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
int main(void)
{
	char str[20];

	printf("단어를 입력하세요(빈칸없이):");	scanf("%s", str);

	printf("점수는 %d\n", calculatePoint(str));
}

// calculatePoint 함수의 정의
```