

컴프 #3 수업에서 다루어지는 것들: 포인터(Pointer) #2

■ Call-by-reference

- 함수 호출시 인수 전달 방법: 아래의 두 경우의 출력 결과는?

	Call-by-value	Call-by-reference
전달 내용	복사본: 인수의 값만 매개변수로 전달	원본: 인수의 주소가 매개변수로 전달
전달 방법	기본적 방법	포인터 사용
	<pre>void f1(int x) { x = x + 10; } int main(void) { int a = 5; f1(a); printf("%d\n", a); }</pre>	<pre>void f2(int *p) { } int main(void) { int a = 5; printf("%d\n", a); }</pre>

□ Call-by-reference 의 사용에 #1: <일반 변수>의 함수 매개변수 전달

아래의 실행결과는? 프로그램에서 a, b 의 값이 바뀌도록 프로그램을 수정하라. 메모리를 그려보라

```
void swap(int x, int y)
{
    int tmp;
    tmp = x;
    x = y;
    y = tmp;
}
int main(void)
{
    int a = 100, b = 200;
    printf("a=%d b=%d\n", a, b);
    swap( a, b);
    printf("a=%d b=%d\n", a, b);
}
```

□ Call-by-reference 의 사용에 #2: <1차원 배열>의 함수 매개변수 전달

아래는 <프로그래밍논리의이해>에서 다룬 정수형 배열의 함수 매개변수 전달 프로그램이다.

우리는 매개변수의 p 를 마치 배열처럼 다루었지만 **사실 p 는 포인터였다!**
p 를 배열적 표현 대신 포인터적 표현으로 다시 써보자.

원리: 배열의 첫 주소(와 그 크기)를 전달함으로써 원본을 전달
이점: 크기가 큰 배열은 복사본을 전달하는 것보다 원본 전달이 효율적
기능: 함수에서 (포인터를 사용하여 변경하면) 원본을 바꾸게 된다

```
void changeData(int p[], int size) // 포인터 표현?
{
    int i;
    for (i = 0; i < size; i++)
        p[i] = 99; // 포인터 표현?
}
void printData(int p[], int size) // 포인터 표현:
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", p[i]); // 포인터 표현?
    printf("\n");
}
int main(void)
{
    int data[5] = {1, 2, 3, 4, 5};

    changeData(data, 5); // data 는 어떤 의미?
    printData(data, 5);
}
```

□ Call-by-reference 의 사용에 #3: <문자열>의 함수 매개변수 전달

아래의 프로그램을 메모리를 그리며 따라가보라.

```
void stringCopy(char *s1, const char *s2)// *s2 는 바뀌지않으므로 const 를 붙일수
{
    while (*s2) // *s2 != '\0', '\0 의 아스키값은 0 이다
    {
        *s1= *s2;
        s1++;
        s2++;
    }
    *s1 = '\0'; // 반드시 문자열의 끝을 null 문자로 채워야
}

int main(void)
{
    char copied[10];
    char data[] = "abc";

    stringCopy(copied, data);
}
```

아래의 두 경우에서 const의 쓰임을 이해하자. 어떤 의미인가?

```
const int * ptr=&num;
int * const ptr=&num;
```

□ Call-by-reference 의 사용에 #4: <2차원 배열>의 함수 매개변수 전달

2차원 배열의 함수 매개변수 전달도 <프로그래밍논리의 이해> 수업에서 다루었다.

여기서도 **pa 는 포인터이다!!**
그러나 아래의 표현처럼 pa[i][j]로 생각하고 코딩을 해보자.

(자세한 것은 2학년의 문제해결기법 시간에 다룬다)

```
void print_array(int pa[][3], int size) //print_array(int (* pa)[3], int size)와 같음
{
    int i, j;
    for (i = 0; i < size; i++)
        for (j = 0; j < 3; j++)
            printf("%d ", pa[i][j]);
}

int main(void)
{
    int array[][3] = {{1, 22, 70}, {23, 80, 34}, {90, 20, 4}, {40, 99, 30}};
    int rowNb = sizeof(array) / sizeof(array[0]);

    print_array(array, rowNb);
}
```

LAB 8 포인터(2)

- 배열의 함수 매개변수 전달
- 문자열의 함수 매개변수 전달

■ LAB8.1

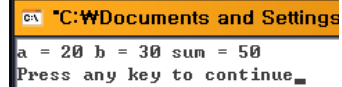
Call-by-value와 Call-by-reference에 대한 연습이다. 아래의 프로그램을 보고 결과를 예측하시오.

```
#include <stdio.h>

void add_two(int x, int y, int s)
{
    s = x + y;
}

void main()
{
    int a = 20, b = 30;
    int sum = 0;

    add_two(a, b, sum);
    printf("a = %d b = %d sum = %d\n", a, b, sum);
}
```



```
C:\Windows and Settings
a = 20 b = 30 sum = 50
Press any key to continue.
```

위의 프로그램을 수정하여 위와 같은 결과가 나오도록 하라. 즉, add_two() 함수는 앞의 두 매개변수를 더해 세 번째 매개변수에 저장하는 함수이다.

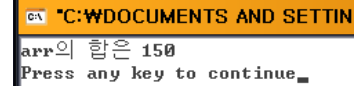
■ LAB8.2 배열의 함수 매개변수 전달에 관한 문제는 1학기에 다루었다.

아래의 프로그램을 살펴보자.

```
#include <stdio.h>

//version 1
int sumUp(int pa[], int size) // B)
{
    int i, sum = 0;
    for (i = 0; i < size; i++)
        sum += pa[i];
    return sum;
}

int main(void)
{
    int arr[5] = {10, 20, 30, 40, 50};
    printf("arr의 합은 %d\n", sumUp(arr, 5)); // A) 배열이름과 그 크기를 인수로 호출한다!
                                           // arr = &arr[0]
}
```



```
C:\Windows and Settings
arr의 합은 150
Press any key to continue.
```

A)에서 arr은 배열의 이름이고 이는 &arr[0]과 같다.

B)에서 pa는 사실은 배열의 이름이 아니고 포인터이다.

위의 sumUp함수는 포인터인 pa의 연산으로 다음과 같이 정의할 수도 있다.

Version 1, version 2, version 3 모두 맞다. 어느 것이 더 이해하기(작성하기) 쉬운가?

(7주차에 <포인터와 배열>에서도 위의 경우를 3가지의 예제로 다루었음을 상기하라)

Version 1: 예제 D

Version 2: 예제 C

Version 3: 예제 B

// version 2

```
int sumUp(int *pa, int size)
{
    int i, sum = 0;
    for (i = 0; i < size; i++)
        sum += *(pa + i); // pa[i]
    return sum;
}
```

//version 3 : pa 값을 변화시키면서!

```
int sumUp(int *pa, int size)
{
    int i, sum = 0;
    for (i = 0; i < size; i++)
        sum += *pa++;
    return sum;
}
```

■ LAB8.3(문자열을 함수의 매개변수로 전달)

□ LAB8.3_1아래 코드는 수업시간에 배운 copy() 함수이다.

즉, copy(char *s1, char *s2) 함수는 s2 문자열의 내용을 s1에 복사하는 함수이다.

(여기서는 s1, s2의 값을 변화시키면서(LAB8_2의 Version 3처럼) 코드를 작성했다)

디버거 수행을 통해(F10과 F11의 적절한 사용) 문자열이 어떻게 복사되는지 확인해 보자. 이때, VS에 나타나는 s1 과 s2의 값 변화에 주의하면서 한 라인씩 수행해 본다. (참고, s1 과 s2가 가리키는 문자의 값을 알고 싶다면, 조사식 창에 *s1, *s2를 입력해 두면 해당 포인터들이 가리키는 문자를 볼 수 있다)

```
// 디버깅 연습
void stringCopy(char *s1, const char *s2)// *s2는 바뀌지않으므로 const를 붙일수
{
    while (*s2)
    {
        *s1= *s2;
        s1++;
        s2++;
    }
    *s1 = '\0';
}

int main(void)
{
    char copied[500];
    char data1[] = "abcde";
    char data2[] = "ABCDEFGH";

    stringCopy(copied, data1);
    stringCopy(copied, data2);
}
```

1) while (*s2)

s2++;

2) while (*s2++);

위의 1)과 2)는 같은가??

답: _____

위의 프로그램에 문자열의 길이를 세는 함수 strlen을 추가 정의(LAB8_2의 Version 3처럼)하고, 이를 main함수에서 호출하여 아래와 같은 결과가 나오게하라. 수정된 main함수는 다음과 같다.

```
// strlen 함수 정의 추가

// stringCopy 함수 정의 - 위의 것 그대로

int main(void)
{
    char copied[500];
    char data1[] = "abcde";
    char data2[] = "ABCDEFGH";

    stringCopy(copied, data1);
    printf("첫번째 복사후 copied의 길이와 값은 %d와 %s이다.\n", strlen(copied), copied);
    stringCopy(copied, data2);
    printf("두번째 복사후 copied의 길이와 값은 %d와 %s이다.\n", strlen(copied), copied);
}
```

C:\WINDOWS\system32\cmd.exe

첫번째 복사를 시행한후 copied의 길이와 값은 5와 abcde이다.
두번째 복사를 시행한후 copied의 길이와 값은 8와 ABCDEFGH이다.
계속하려면 아무 키나 누르십시오 . . .

□ LAB8.3.2 (문자열을 함수의 매개변수로 전달하는 함수를 한가지 더 연습해보자. 두개의 문자열을 비교해서 같으면 0을 다르면 1을 반환하는 함수 strcmp을 정의하여(LAB8_2의 Version 3처럼) 아래의 실행결과가 나오게하라.

C:\WINDOWS\system32\cmd.exe

Enter the first string:Suehee
Enter the second string:Sue
두개의 문자열은 다르다
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe

Enter the first string:Suehee
Enter the second string:Suehee
두개의 문자열은 같다
계속하려면 아무 키나 누르십시오 . . .

```
int strcmp(const char *s1, const char *s2)
{
    // 정의 추가
}

#include <stdio.h>
void main()
{
    char string1[50];
    char string2[50];

    printf("Enter the first string:");
    scanf("%s", string1);
    printf("Enter the second string:");
    scanf("%s", string2);

    if (strcmp(string1, string2) == 0)
        printf("두개의 문자열은 같다\n");
    else
        printf("두개의 문자열은 다르다\n");
}
```

- **LAB8.4(2차원 배열을 함수의 매개변수로 전달)** 아래의 코드는 2차원 배열 array의 엘리먼트들을 출력하고 그 들중 가장 큰수를 구하여 출력하는 프로그램이다.

main함수에서 rowNb값을 예측해보라.

int rowNb = sizeof(array) / sizeof(array[0]);

함수 print_array에서 매개변수인 pa를 표현한 것처럼(LAB8.2의 Version 2처럼)

max_array도 프로그램을 정의하라.

(이에 대한 포인터적 표현 및 사용은 2학년에서 배운다)

```
#include <stdio.h>
void print_array(int pa[][3], int size) {
    int i, j;
    for (i = 0; i < size; i++)
        for (j = 0; j < 3; j++)
            printf("%d ", *(pa[i]+ j))
}

int max_array(int pa[][3], int size)
{
    int i, j, max;
    // 여기에 코드 삽입

}

int main(void)
{
    int array[][3] = {{1, 22, 70}, {23, 80, 34}, {90, 20, 4}, {40, 99, 30}};
    int rowNb = sizeof(array) / sizeof(array[0]);

    print_array(array, rowNb);
    printf("\n가장 큰 값은 %d\n", max_array(array, rowNb));
}
```

HW 8 포인터(2)

- **HW8.1 배열의 함수 매개변수 전달(난이도 중)**

아래의 실행결과가 나오도록 프로그램을 완성하라. main함수에는 추가할 필요가 없다. 나머지 세 개의 함수의 정의부분을 완성하면 된다. 난수발생을 위한 seed로 현재의 시간을 이용하는 time()함수를 사용했으므로 실행결과란 매번 다른 수들을 보여줄 것이다.

요구사항: 아래의 함수들을 정의할 때 위의 LAB8.2에서의 version3처럼 작성하라.

```
C:\Windows and Settings\Wuser\My Docu
엘리먼트의 합은 482
엘리먼트들은 54 60 69 28 54 17 65 17 50 68
엘리먼트들 중 가장 큰수는 69
Press any key to continue.
```

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

```
int sumUpData(int *, int);
void printData(int *, int);
int maxData(int *, int);
```

```
void main()
{
```

```
    int k;
    int *p;
    int sum;
```

```
    int data[10]; // 계산 수행에 사용할 정적 변수
```

```
    p = data; //혹은 p = &data[0]
    srand(time(NULL)); // random 값 출력에 사용하는 함수. Seed 값을 부여
```

```
    for (k = 0; k < 10; k++) // Index 0..9까지
        *p++ = rand() % 100; // 데이터 초기화. 0부터 99까지의 Random 값을 출력.
```

```
    sum = sumUpData(data, 10);
    printf("엘리먼트의 합은 %d\n", sum);
```

```
    printf("엘리먼트들은 ");
    printData(data, 10);
```

```
    printf("\n엘리먼트들 중 가장 큰수는 %d \n", maxData(data, 10));
}
```

```
int sumUpData(int *p, int size) // int p[]로 써도 된다
{
```

```
}
```

```
void printData(int *p, int size) // int p[]로 써도 된다
{
```

```
}
```

```
int maxData(int *p, int size) // int p[]로 써도 된다
{
```

```
}
```

■ HW8.2 (문자열을 함수의 매개변수로 전달)

아래와 같은 기능을 하는 프로그램을 작성하라.

- 이 두개의 이름이 같은가, 혹은 알파벳으로 앞서는가 뒤서는가(즉, 정렬되어있는가)를 판별하여 출력하고
- 이 이름을 붙여서 출력하고,
- 이 이름 안의 'u' 글자를 'x' 로 바꾸어서 출력한다.

아래의 함수들을 LAB8_2의 Version 3처럼 포인터 값을 변화시키면서 작성하라.

- **stringCompare**(s1, s2)함수는 s1과 s2 문자열이 같으면 0을, 두 문자열이 정렬되었으면 1을, 정렬되어있지않으면 -1을 반환하는 함수이다. 힌트: LAB8_3_2의 stringEqual을 조금 변형하라.

정렬의 의미는? 답_____

- **stringCat**(s1, s2)함수는 s2 문자열을 s1 끝에 접합하는 함수이다.
- **stringChange**(s, ch, newCh)함수는 s 문자열 안의 ch 문자를 newCh로 바꾸는 함수이다.

```
#include <stdio.h>

int stringCompare(const char *s1, const char *s2)
{
    // 정의 작성
}

void stringCat(char *s1, const char *s2)
{
    // 정의 작성
}

void stringChange(char *s, char ch, char newCh)
{
    // 정의 작성
}

int main(void)
{
    char name1[20];
    char name2[20];

    printf("Enter the first name: ");
    scanf("%s", name1);
    printf("Enter the second name: ");
    scanf("%s", name2);

    if (stringCompare(name1, name2) == 0)
        printf("두개의 이름은 같다\n");
    else if (stringCompare(name1, name2) == 1)
        printf("두개의 이름은 다르며 정렬되어있다\n");
    else
        printf("두개의 이름은 다르며 정렬되어있지않다\n");

    stringCat(name1, name2);
    printf("The concatenated name is %s\n", name1);

    stringChange(name1, 'u', 'x');
    printf("The changed name is %s\n", name1);
}
```

- **HW8.3 (2차원배열을 함수의 매개변수로 전달)** LAB8_4a에서의 2차원배열 array의 엘리먼트들의 합을 구하는 함수 sum_array()를 추가하여 아래와 같은 실행결과가 나오도록 프로그램을 작성하라.

여기서는

LAB8_2의 Version 2처럼 작성해본다.

int sum_array(int pa[][3], int size) // sum_array(int (* pa)[3], int size)와 같음

```
C:\ *C:\Documents and Settings\Wuser\W...
1 22 70 23 80 34 90 20 4 40 99 30
가장 큰 값은 99
엘리먼트들의 합은 513
Press any key to continue_
```

```
C:\ C:\WINDOWS\system32\cmd.exe
Enter the first name: Suehee
Enter the second name: Suehee
두개의 이름은 같다
The concatenated name is SueheeSuehee
The changed name is SxeheeSxehee
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\ C:\WINDOWS\system32\cmd.exe
Enter the first name: Abba
Enter the second name: Suehee
두개의 이름은 다르며 정렬되어있다
The concatenated name is AbbaSuehee
The changed name is AbbaSxehee
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\ C:\WINDOWS\system32\cmd.exe
Enter the first name: Suehee
Enter the second name: Abba
두개의 이름은 다르며 정렬되어있지않다
The concatenated name is SueheeAbba
The changed name is SxeheeAbba
계속하려면 아무 키나 누르십시오 . . .
```