

## 자료구조 보충자료: 순환

- 순환 연습1: 다음과 같은 실행결과를 내도록 재귀적으로 함수 fact를 완성하라.

```
Enter a number:5
5! = 120
계속하려면 아무 키나 누르십시오 . .
```

```
Enter a number:0
0! = 1
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int fact(int n)
{
```

```
}
```

```
int main(void)
{
    int n;
    printf("Enter a number:");
    scanf("%d", &n);

    printf("%d! = ", n);
    printf("%d\n", fact(n));
}
```

- 순환 연습2: 다음과 같은 실행결과를 내도록 재귀적으로 함수 print\_fact\_exp를 완성하라

```
Enter a number:5
5! = 5 * 4 * 3 * 2 * 1
계속하려면 아무 키나 누르십시오 . . .
```

```
Enter a number:0
0! = 1
계속하려면 아무 키나 누르십시오 . . .
```

```
Enter a number:1
1! = 1
계속하려면 아무 키나 누르십시오 . . .
```

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

```
void print_fact_exp(int n)
{
```

```
}
```

```
int main(void)
{
    int n;
    printf("Enter a number:");
    scanf("%d", &n);

    printf("%d! = ", n);
    print_fact_exp(n);
    printf("\n");
}
```

## REVIEW 1 : 순환

### ■ Review1\_0

### ■ Review1\_1

- ☐ **Review1\_1\_0(순환 연습)** 사용자가 입력한 문자열을 역순으로 출력하는 프로그램을 순환을 이용하여 완성하시오.

```
#include <stdio.h>
#include <string.h>
void print_reverse(char *str, int end)
{
    ...

}

int main()
{
    char str[100];

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

    printf("Reversed String is: ");
    print_reverse(str, strlen(str) - 1); // str과 마지막 인덱스를 매개변수로
    return 0;
}
```

실행 결과(예):

```
Enter any string: recursion
Reversed String is: noisrucer
```

- ☐ **Review1\_1\_1(순환 연습)** 사용자가 입력한 문자열의 역문자열을 만드는 프로그램을 순환(recursion)을 이용하여 작성하시오. (단, 입력 문자열 str[] 외의 다른 배열은 사용하지면 안 됨) 프로그램의 main() 함수는 다음과 같다. 작성한 프로그램을 실행해 보시오.

```
#include <stdio.h>
#include <string.h>
void reverse_string(char *str, int s, int e)
{
    ...// 작성할 것
}

int main()
{
    char str[100];

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

    reverse_string(str, 0, strlen(str) - 1);
    printf("\nReversed String is: %s", str);
    return 0;
}
```

- ☐ **Review1\_1\_2(반복 연습)** 위 문제를 순환 대신 반복(iteration)을 이용하여 작성해 보시오.(main함수는 위의 Review1\_1과 그대로)

## 자료구조 보충자료: 구조체

### #1 구조체 기본

#### 방법1

- 구조체명 선언(정의)

```
struct person {
    char name[10];
    int age;
}; // person을 구조체명
```

- 구조체변수 선언(구조체명 이용)

```
struct person p; // p는 구조체변수. 메모리가 할당된다.
```

Q: 이때 sizeof(struct person)의 값은? **16 byte**

#### 방법2: typedef를 사용

- 구조체타입 선언(정의)

```
typedef struct person { // 이 시점에서 person은 구조체명, 여기서는 person은 생략 가능
    char name[10];
    int age;
} person; // 이 시점 이후 person은 구조체타입.
```

- 구조체변수 선언(구조체타입 이용)

```
person p;
```

Q: 어느것이 더 better? **typedef**

Typedef를 사용한 방법2를 이용하여 구조체타입 person을 정의, 그것에 대한 구조체변수 p를 위처럼 선언하였다고 가정하자.

- 구조체를 이용한 각종 활동들

- 구조체변수의 구조체 항목 접근

**p.name p.age**

- person을 가르키는 포인터 ptr 선언, 구조체 변수 p를 포인터 ptr이 가리키게

```
Person *ptr // 4바이트의 포인터 ptr가 생성
ptr = &p
```

- 구조체 포인터의 구조체 항목 접근

**ptr -> name ptr -> age**

- person타입의 메모리를 생성하여 ptr이 가르키게하는 문장

```
ptr = (person *)malloc(sizeof(person));
```

- malloc 함수: 주어진 메모리크기만큼의 메모리를 생성(할당)하여 그것을 가리키는 포인터를 반환한다

#### 형식

```
(타입 *)malloc(메모리크기)
```

- 타입: int, 혹은 person 등등

- 메모리 크기(바이트 단위): 4, 100, sizeof(int), sizeof(person) 등등

#### 사용예

```
char *pc;
int *pi;
person *ptr; // 항상 포인터는 4byte
```

```
pc = (char *)malloc(sizeof(char)); // char 형 메모리를 생성하여 pc 이 가르키게
```

```
pi = (int *)malloc(sizeof(int)); // int 형 메모리를 생성하여 pi 이 가르키게
```

```
ptr = (person *)malloc(sizeof(person)); // person 타입의 메모리를 생성하여 ptr 이 가르키게
```

### #2 구조체 연습

#### //컴프2 LAB10\_0.0

```
struct student {
    char name[10];
    int midterm;
    int final;
};
#include <stdio.h>
int main(void)
{
    struct student aStudent;

    printf("Enter student name: ");
    scanf("%s", aStudent.name); //1)
    printf("Enter midterm and final score: ");
    scanf("%d %d", &aStudent.midterm, &aStudent.final); //2)

    printf("\n이름\t중간\t학기말\n");
    printf("%s\t%d\t%d\n", aStudent.name, aStudent.midterm, aStudent.final); //3)
}
```

#### //컴프2 LAB10\_0.2

```
struct student {
    char name[20];
    int midterm;
    int final;
};
#include <stdio.h>
int main(void)
{
    struct student aStudent;
    struct student *sp = &aStudent; //4) sp가 aStudent를 가르키게한다

    printf("Enter student name: ");
    scanf("%s", sp->name); //5)
    printf("Enter midterm and final score: ");
    scanf("%d %d", &sp->midterm, &sp->final); //6)

    printf("\n이름\t중간\t학기말\n");
    printf("%s\t%d\t%d\n", sp->name, sp->midterm, sp->final); //7)
}
```

- 다음의 순서로 프로그램을 완성,수정하라.

1. 위의 빈칸을 채워라

2. 컴프LAB10\_0\_2를

- (A. typedef를 이용하여 Student 타입을 정의하여 프로그램을 수정하라.

- B. 포인터 sp만을 사용하여 프로그램하라. aStudent를 사용하지 않으면 어떻게 해야하는가?

- ① sizeof(Student)의 값은?

- ② Student 타입의 메모리를 생성하여 sp가 가르키게 하는 문장은?

구조체

```
// Review2_1
#include <stdio.h>
#include <stdlib.h>
// 1) typedef으로 구조체 타입 Score 정의
```

```
void printScore(Score * p) //2)
{
    printf("중간고사 성적은 %d\n",          );
    printf("학기말고사 성적은 %d\n",      );
}
Score* biggerScore(Score *p1, Score *p2) //3)
{
```

```

}
Score* totalScore(Score *p1, Score *p2) //4)
{
```

```

}
Score *createScore(int m, int f) //5)
{
```

```
// Review2_1_1
int main(void) {
    Score *p1, *p2, *p3;
    //1)p1, p2에 Score를 저장하기위한 메모리를 할당하고 값을 대입한다. 각각 (50, 100), (70, 70)

    printScore(p1);
    printScore(p2);

    printf("-----\n");
    printf("둘중 성적이 좋은 점수:\n");
    printScore(biggerScore(p1, p2));

    printf("-----\n");
    printf("두 성적의 총 합:\n");
    printScore(totalScore(p1, p2));

    printf("-----\n");
    p3 = createScore(100, 100);
    printScore(p3);
}
```

```
// Review 2_1_2
int main(void) {
    Score s1, s2; // 포인터 p1, p2대신에 구조체변수 s1,s2
    Score *p3;

    printScore(&s1);
    printScore(&s2);

    printf("-----\n");
    printf("둘중 성적이 좋은 점수:\n");
    printScore(          );

    printf("-----\n");
    printf("두 성적의 총 합:\n");
    printScore(          );

    printf("-----\n");
    p3 = createScore(100, 100);
    printScore(p3);
}
```

## REVIEW 2 : 배열, 구조체, 포인터

### ■ Review2\_0

3장 배열, 구조체, 포인터의 Quiz/연습문제 중 일부

### ■ Review2\_1(포인터와 구조체)

#### □ Review2\_1\_1(구조체 포인터 연습)

아래와 같은 구성요소를 갖도록 typedef를 사용하여 구조체 타입 Score를 선언하라.

- ```
int midterm;
int final;
```
- 1) main 함수 안에서 이 Score 타입의 변수를 가리키는 포인터 p1, p2를 선언하고, 동적 메모리 할당을 한 후 p1이 가리키는 점수에 중간고사 50, 학기말고사 100을 p2가 가리키는 점수에 중간고사 70, 학기말고사 70을 대입.
  - 2) 하나의 Score 포인터가 가리키는 중간고사, 학기말고사 성적을 아래와 같이 출력하는 함수 void printScore(Score \*p)를 작성.  
중간고사 성적은 \*\*  
학기말고사 성적은 \*\*
  - 3) 두 개의 Score중 성적의 합이 큰 Score를 반환하는 함수 Score \*biggerGrade(Score\* p1, Score\* p2)를 작성.
  - 4) 두 개의 Score를 각각(중간고사끼리, 학기말고사끼리)의 점수를 더한 Score 변수에 대한 포인터를 반환하는 Score \*totalScore(Score \*p1, Score \*p2)를 작성.
  - 5) 중간고사와 학기말 고사를 매개변수로 받아서 이를 필드 값으로 갖는 Score 변수에 대한 포인터를 반환하는 함수 Score \*createScore(int m, int f)를 작성.

```
#include <stdio.h>
#include <stdlib.h>
//1) typedef으로 Score를 정의

void printScore(Score * p) {...} //2)
Score *biggerScore(Score *p1, Score *p2) {...} //3)
Score *totalScore(Score *p1, Score *p2) {...} //4)
Score *createScore(int m, int f) {...} //5)
int main(void) {
    Score *p1, *p2, *p3;
    //1) p1, p2를 각각 메모리 할당을 한 후
    // 가리키는 변수에 값 대입

    printScore(p1);
    printScore(p2);

    printf("-----\n");
    printf("둘중 성적이 좋은 점수:\n");
    printScore(biggerScore(p1, p2));

    printf("-----\n");
    printf("두 성적의 총 합:\n");
    printScore(totalScore(p1, p2));

    printf("-----\n");
    p3 = createScore(100, 100);
    printScore(p3);
}
```

```
C:\Windows\system32\cmd.exe
중간고사 성적은 50
학기말고사 성적은 100
중간고사 성적은 70
학기말고사 성적은 70
-----
둘중 성적이 좋은 점수:
중간고사 성적은 50
학기말고사 성적은 100
-----
두 성적의 총 합:
중간고사 성적은 120
학기말고사 성적은 170
-----
중간고사 성적은 100
학기말고사 성적은 100
계속하려면 아무 키나 누르십시오 . . .
```

#### □ Review2\_1\_2(구조체 변수 연습)

위의 프로그램에서 main함수를 다음과 같이 변경하여 같은 실행결과가 나오게 하라.

```
int main(void) {
    Score s1, s2; // 포인터 p1, p2대신에 구조체변수 s1, s2를 사용한다
    Score *p3;

    ...}
```