

자료구조응용

02. 함수의 값 전달, 파일입출력 (10점)

2020.3.18 (수)

1. 다음 프로그램에서 필요한 함수(swap1, swap2, swap3)를 정의하고, 각 함수가 호출된 후 반환 직전의 스택 상태를 그려라. (2점)

[main 함수]

```
int main(void)
{
    int a, b;

    int *pa = &a;
    int **ppa = &pa;

    int *pb = &b;
    int **ppb = &pb;

    a = 1; b = 2; swap1(a, b);      // call by value -- (1)
    printf("After calling swap1(a, b), a: %d, b: %d\n\n", a, b);

    a = 1; b = 2; swap2(&a, &b);    // call by (address) value -- (2)
    printf("After calling swap2(&a, &b), a: %d, b: %d\n\n", a, b);

    a = 1; b = 2; swap2(pa, pb);    // call by (address) value -- (3)
    printf("After calling swap2(pa, pb), a: %d, b: %d\n\n", a, b);

    a = 1; b = 2; swap3(&pa, &pb); // call by (address) value -- (4)
    printf("After calling swap3(&pa, &pb), a: %d, b: %d\n\n", a, b);

    a = 1; b = 2; swap3(ppa, ppb); // call by (address) value -- (5)
    printf("After calling swap3(ppa, ppb), a: %d, b: %d\n\n", a, b);

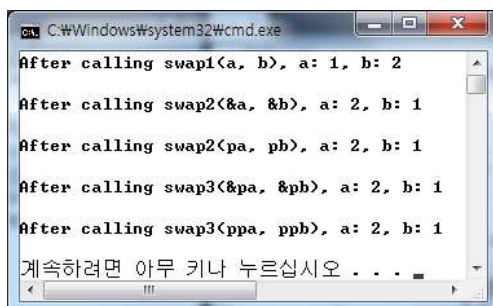
    return 0;
}
```

[함수설명]

swap1 - 함수 내부에서만 두 매개 변수 값의 교환이 이루어짐

swap2, swap3 - 함수 호출로 인해 main의 a, b 값이 교환됨

[실행결과]



```
C:\Windows\system32\cmd.exe
After calling swap1(a, b), a: 1, b: 2
After calling swap2(&a, &b), a: 2, b: 1
After calling swap2(pa, pb), a: 2, b: 1
After calling swap3(&pa, &pb), a: 2, b: 1
After calling swap3(ppa, ppb), a: 2, b: 1
계속하려면 아무 키나 누르십시오 . . .
```

(0) swap1, swap2, swap3의 함수 정의문을 기술하라.

(1) swap1 반환직전 스택상황(변수, 변수값 등)을 그리시오.

(2) swap2 반환직전 스택상황을 그리시오.

(3) swap2 반환직전 스택상황을 그리시오.

(4) swap3 반환직전 스택상황을 그리시오.

(5) swap3 반환직전 스택상황을 그리시오.

※ (0)~(5)를 연습장에 직접 작성한 후, 이미지를 찍어 한글 파일에 삽입/수정하여 보고서에 추가할 것. 포인터에 의한 참조는 화살표로 표현할 수 있음

2. 다음 각 프로그램을 구현 해 본 후 메모리상태(스택 & 힙)를 그려서 보고서에 추가하라.
(4점)

(1) ①, ②, ③ 문장이 실행된 직후의 메모리 상태(스택 & 힙)를 각각 그리기

self check: ②문장이 없다면 ③ 위치 혹은 프로그램이 끝난 뒤 어떤 상황이 될까?

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

void func(int num)
{
    int *pNum = (int *)malloc(sizeof(int));
    *pNum = num;
    printf("%d\n", *pNum); //①
    free(pNum);           //②
}

int main(void)
{
    int num = 1;
    func(num);
    printf("%d\n", num); //③

    return 0;
}
```

(2) ①, ②, ③ 문장이 실행된 이후의 메모리 상태(스택 & 힙)를 각각 그리기
 self check: ③문장이 없다면 프로그램이 끝난 뒤 어떤 상황이 될까?

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

int *func(int num)
{
    int *pNum = (int *)malloc(sizeof(int));
    *pNum = num;
    printf("%d\n", *pNum);    //①
    return pNum;
}

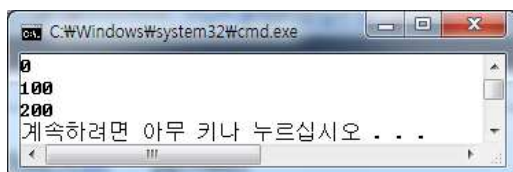
int main(void)
{
    int num = 1;
    int *pNum = func(num);
    printf("%d\n", num);      //②
    free(pNum);               //③

    return 0;
}
```

3. 다음 프로그램의 func1, func2, func3을 정의한 후 실행되도록 구현하라. 그리고, func1, func2, func3, main 함수의 종료직전 메모리 상태(스택 & 힙)를 각각 그려서 보고서에 추가하라.(2점)

```
int main(void)
{
    int *pNum = func1(); // int 크기 만큼 동적할당 받아 0을 지정
    printf("%d\n", *pNum);
    printf("%d\n", *func2( pNum )); // pNum이 가리키는 것을 100으로 수정 후 출력
    printf("%d\n", **func3( &pNum )); // pNum이 가리키는 것을 200으로 수정 후 출력
    free(pNum);

    return 0;
}
```



4. 다음과 같은 파일 입출력 프로그램을 작성하라.(2점)

[main 함수]

```
int main(void)
{
    int num, *pNum;

    FILE *fpIn = fopen("input.txt", "r");
    FILE *fpOut = fopen("output.txt", "w");

    fscanf(fpIn, "%d", &num); // 데이터 갯수
    pNum = alloc(num);        // num 크기 만큼 메모리 할당
    input(pNum, num, fpIn);   // 파일로부터 num개 데이터 입력
    increase(pNum, num);      // 각 데이터를 1씩 증가
    output(pNum, num, fpOut); // 증가된 데이터를 파일출력

    free(pNum);
    fclose(fpIn);
    fclose(fpOut);

    return 0;
}
```

[입출력 파일형식]

input.txt	output.txt
10	10
1 2 3 4 5 6 7 8 9 10	2 3 4 5 6 7 8 9 10 11

■ 제출 형식

- 솔루션 이름 : DS 02
- 프로젝트 이름 : 1, 2-1, 2-2, 3, 4
- 각 소스파일에 주석처리
“학번 이름”
“본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”
- 실행화면을 캡처하여 한글파일에 추가 후 솔루션 폴더에 포함
- 솔루션 정리 메뉴를 수행 후 전체 솔루션을 “학번.zip”으로 압축하여 과제 게시판에 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!
- 마감 : 다음 수업일 전 (3월 22일, 일요일) 자정