

프로그래밍 연습

실습 week 8

과제 풀이

실습

실습1

재귀 함수를 이용하여 0~n까지 합산하는 프로그램

```
#include <stdio.h>

int mycal(int n);

int main(void)
{
    int num=0;

    printf("n을 입력하세요 : ");
    scanf("%d", &num);
    printf("\n결과값은 %d 입니다.\n", mycal(num));

    return 0;
}
```

실습1

재귀 함수를 이용하여 0~n까지 합산하는 프로그램

```
int mycal(int n)
{
    static int enter_count=1;
    int count=0, ret=0, i=0;

    count = enter_count++;

    for(i=0;i<count-1;i++) printf(" "); // 현재 recursion depth 만큼 들여쓰기
    printf("mycal() is called %d times. (argument : %d)\n", count, n);

    if(n <= 1) {
        ret = 1;
        for(i=0;i<count-1;i++) printf(" "); // 현재 recursion depth 만큼 들여쓰기
        printf("mysql() exit with return value %d (= %d)\n", ret, n);
    } else {
        ret = n + mycal(n-1);
        for(i=0;i<count-1;i++) printf(" "); // 현재 recursion depth 만큼 들여쓰기
        printf("mysql() exit with return value %d (= %d + %d)\n", ret, ret-n, n);
    }

    return ret;
}
```

[실행 결과]

```
n을 입력하세요 : 10
mycal() is called 1 times. (argument : 10)
  mycal() is called 2 times. (argument : 9)
    mycal() is called 3 times. (argument : 8)
      mycal() is called 4 times. (argument : 7)
        mycal() is called 5 times. (argument : 6)
          mycal() is called 6 times. (argument : 5)
            mycal() is called 7 times. (argument : 4)
              mycal() is called 8 times. (argument : 3)
                mycal() is called 9 times. (argument : 2)
                  mycal() is called 10 times. (argument : 1)
                    mysql() exit with return value 1 (= 1)
                      mysql() exit with return value 3 (= 1 + 2)
                        mysql() exit with return value 6 (= 3 + 3)
                          mysql() exit with return value 10 (= 6 + 4)
                            mysql() exit with return value 15 (= 10 + 5)
                              mysql() exit with return value 21 (= 15 + 6)
                                mysql() exit with return value 28 (= 21 + 7)
                                  mysql() exit with return value 36 (= 28 + 8)
                                    mysql() exit with return value 45 (= 36 + 9)
                                      mysql() exit with return value 55 (= 45 + 10)
```

결과값은 55 입니다.

실습1

➤ 코드분석

```
static int enter_count=1;
```

- mycal() 함수 진입 횟수 세기 위한 목적의 변수.
함수 재진입하더라도 static 이라서 초기화되지 않고,
이전 값을 유지하고 있다.

```
for(i=0;i<count-1;i++) printf(" ");
```

- 호출된 mycal() 함수 recursion depth 만큼 들여쓰기를 한다.

```
printf("mycal() is called %d times. (argument : %d)\n", count, n);
```

- 현재 mycal() 함수 recursion depth와 넘겨받은 argument를 출력한다.

실습1

➤ 코드분석

```
if(n <= 1) {  
    ret = 1;  
    for(i=0;i<count-1;i++) printf(" ");  
    printf("mysql() exit with return value %d (= %d)\n", ret, n);  
}
```

- 넘겨받은 argument가 1보다 작거나 같을 경우 recursion을 중단하고 return할 변수에 1을 할당한다.
- mycal() 함수 진입 횟수 만큼 들여쓰기 후에 넘겨 줄 값을 출력한다.

실습1

➤ 코드분석

```
else {  
    ret = n + mycal(n-1); for(i=0;i<count-1;i++) printf(" ");  
    printf("mysql() exit with return value %d (= %d + %d)\n", ret, ret-n, n);  
}
```

- 넘겨받은 argument가 1보다 클 경우 n-1을 argument로 recursion을 수행한다. Recursion 결과는 n 과 더하여 준다.
- mycal() 함수 진입 횟수 만큼 들여쓰기 후에 recursion을 통해 구해진 값과 현재 argument n을 출력한다.

실습 2

문제 피보나치 수 구하기

0번째 피보나치 수는 0, 1번째 피보나치 수는 1입니다. 이 외 피보나치 수는 본인의 바로 앞 두 수의 합이 됩니다. $F_n = F_{n-1} + F_{n-2}$ ($n \geq 2$)

입력

자연수 n ($1 \leq n \leq 50$)

출력

n 번째 피보나치 수

테스트케이스

입력

1

출력

1

입력

10

출력

55

입력

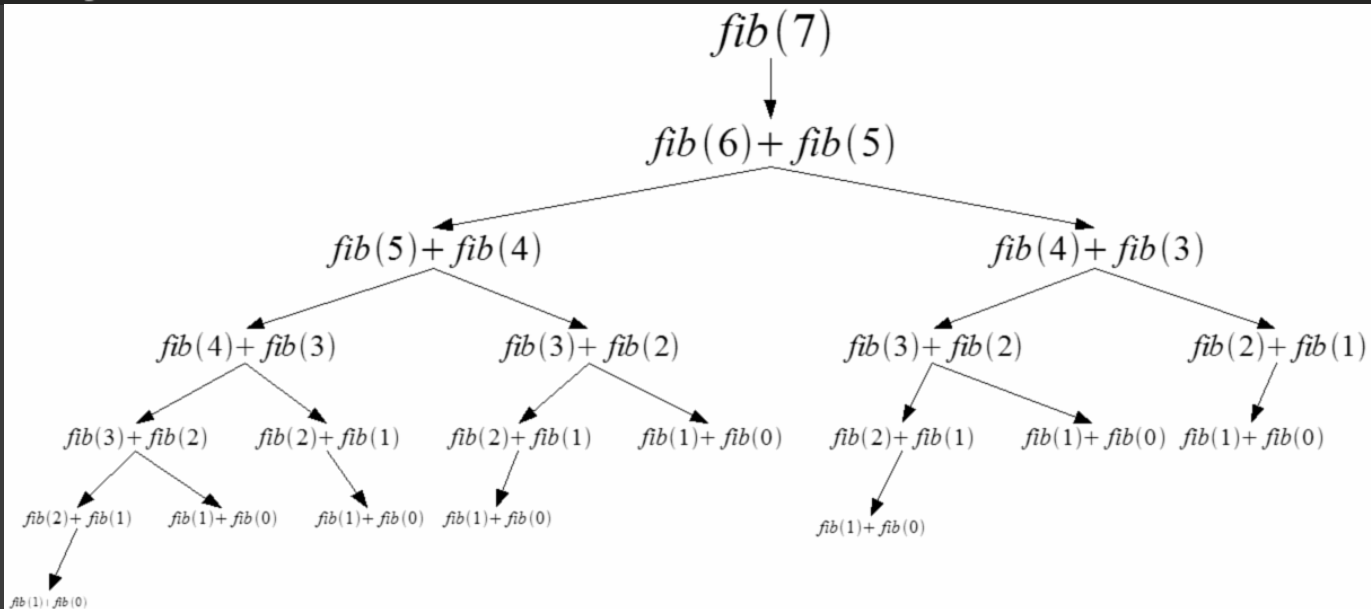
40

출력

102334155

실습 2

```
1  #include <stdio.h>
2
3  long fib(int n);
4
5  int main(){
6      int n;
7      scanf("%d", &n);
8      printf("%ld\n", fib(n));
9
10     return 0;
11 }
```



실습3

포인터의 이해

```
1  #include <stdio.h>
2
3  void swap1(int a, int b){
4      int t = a;
5      a = b;
6      b = t;
7  }
8
9  void swap2(int* a, int* b){
10     int t = *a;
11     *a = *b;
12     *b = t;
13 }
14
15 int main(){
16     int a = 3;
17     int b = 5;
18     swap1(a, b);
19     printf("%d %d\n", a, b);
20     swap2(&a,&b);
21     printf("%d %d\n", a, b);
22
23     return 0;
24 }
```

실습 3

Swap1 함수는 value로 parameter를 전달하는 call by value
따라서 함수가 종료 후 메인 함수의 a, b의 값은 변화가 없음

출력

3 5

Swap2 함수는 주소로 parameter를 전달하는 call by reference
따라서 함수가 종료 후 메인 함수의 a, b 값도 변화 돼 있음

출력

5 3

실습4

문자열 다루기

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  int main(){
6      char words[5][20] = {"programming", "practice", "computer", "computer",
7                          "123456"};
8
9      printf("length %ld, %ld\n", strlen(words[0]), strlen(words[1]));
10     printf("compare 1 and 2 : %d\n", strcmp(words[1], words[2]));
11     printf("compare 2 and 1 : %d\n", strcmp(words[2], words[1]));
12     printf("compare 2 and 3 : %d\n", strcmp(words[2], words[3]));
13     printf("string to long %ld\n", strtol(words[4], NULL, 10));
14     return 0;
}
```

실습 4

string.h 라이브러리

strlen은 문자열의 길이를 구하여 정수형으로 반환

출력

118

strcmp는 문자열을 비교. strcmp(left, right) 를 호출했을 때 left와 right가 같으면 0, left가 알파벳 순으로 작으면 음수, 크면 양수 반환

출력

13

-13

0

stdlib.h 라이브러리

strtol 은 string을 long형으로 변환 후 반환. atoi, itoa 등이 있음.

출력

123456

과제

과제 유의 사항

어길 시 감점 요인이 될 수 있습니다.

- 파일명 / 폴더명 양식 준수 (공지사항 참고)
- 출력 형식 준수
- 컴파일 에러, 미완성 코드 등은 미제출로 간주

과제 (파일명 : hw1.c)

문제.

주어진 코드를 완성 해 10개의 문자열을 길이순으로, 길이가 같다면 알파벳 순으로, 오름차순 정렬 하세요.

복사 할 수 있는 코드는 수업 당일 공지사항에 업로드 예정입니다.

입력 : 없음

출력 : 정렬 된 문자열을 한 줄에 한 개씩 출력

```
출력
game
help
time
apple
linux
money
banana
window
windows
computer
```

과제 (파일명 : hw1.c)

코드.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void swap(char** , char**);
5  void bubble_sort(char* words[10]);
6
7  int main(void) {
8      char* words[10] = {
9          "computer", "windows", "window", "linux",
10         "apple", "banana", "time", "help",
11         "game", "money"};
12     bubble_sort(words);
13     for(int i = 0; i < 10; i++){
14         printf("%s\n", words[i]);
15     }
16 }
17
18 void swap(char** a, char** b){
19     /****** */
20     /* implement this function */
21 }
22
23 void bubble_sort(char* words[10]){
24     /* implement this function */
25     /****** */
26 }
```