

프로그래밍 연습

2017-10-12

6주차 실습

채점 사이트 링크

<http://pp1.snucse.org:8888>

Topic

- Function
- 이중 For 문

scanf 총정리

```
1  #include <stdio.h>
2
3  int main(void) {
4      int a;
5      long long b;
6      char c[100];
7      // 32bit integer
8      scanf("%d",&a);
9      // 64bit integer
10     scanf("%lld",&b);
11     // string
12     scanf("%s",c);
13     // EOF
14     while(scanf("%d",&a)!=EOF);
15     return 0;
16 }
```

- read *formatted* data from stdin : format과, 변수 목록을 지정해줌
- 공백문자(space, newline, tab)를 자동으로 무시한다
 - ex) scanf("%d%d", &a, &b); 만 해도 두 개 숫자 입력
- format
 - %d : int형 데이터
 - %lld : long long형 데이터
 - %f : float형 데이터
 - %lf : double형 데이터
 - %s : char[]형 데이터
 - and more..

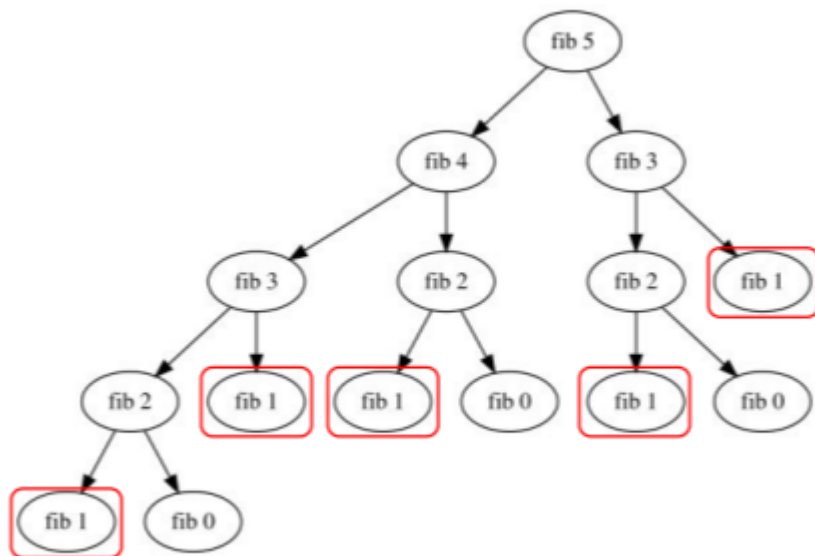
} &를 붙여준다
(printf에는 해당없음!)
- return value : 입력이 끝난 경우 EOF를 리턴
- 참고 : <http://www.cplusplus.com/reference/cstdio/scanf/?kw=scanf>

재귀 호출(Recursion)

- 함수가 스스로를 호출하는 것
- 기본적인 구조
 - 종료조건 : 필수!
 - 계산결과를 얻어와 합치는 과정

```
int fib(int n){  
    if(n==0) return 0;  
    if(n==1) return 1;  
    return fib(n-1) + fib(n-2);  
}
```

- fib(5)를 실행한 결과
 - 같은 값을 반복해서 계산한다
ex) fib(1)은 5번 호출된다
 - 간단한 코드 but 효율성을 잘 따져야



Recursive version to Non-recursive version

- Factorial : recursive version

```
long long fact(int n){  
    if(n==1) return 1;  
    return n*fact(n-1);  
}
```

- Factorial : non-recursive version

```
long long fact[100];  
fact[1] = 1;  
for(int i=2;i<=n;i++){  
    fact[i] = i*fact[i-1];  
}
```

- 함수에서 값을 계산해 **return**하는 대신, 배열에 값을 쌓아나가는 방식
- 이전 값들이 계산되었다는 것이 보장되어야 한다
 - fact[i]를 계산하기 위해서는 fact[i-1]이 먼저 계산되어 있어야 함

이중 For 문

- 중첩 For문 사용법

```
For loop 1{  
    For loop 2{  
        [연산]  
    }  
}
```

Loop1 : n번, Loop2 : m 번 반복 된다고 할 때,
[연산]은 $m * n$ 번 수행

```
for (i = 0; i < 9; i++)  
{  
    for (j = 0; j < 9; j++)  
    {  
        printf("%d * %d = %d\n", i, j, i * j);  
    }  
    printf("\n");  
}
```

HW 1.

Description

Given an integer $N(0 \leq N \leq 50)$, print the N -th Fibonacci number. You SHOULD implement this without recursion. Otherwise, it would cause TLE(time limit exceeded).

Fibonacci number is defined as:

$$f(n) = \begin{cases} n = 0 & 0 \\ n = 1 & 1 \\ n > 1 & f(n-1) + f(n-2) \end{cases}$$

Hint

Use type *'long long'*.

HW 1. 예시

Input	Output
5	5
6	8
50	12586269025

HW 2. Recursion

Description

Given two integers N , K ($0 \leq K \leq N \leq 25$), print the binomial coefficient $C(N, K)$, which is the number of ways to choose K elements, disregarding their order, from a set of N elements.

You can use recursion here.

Hint

You can use the following recurrence relation

$$C(n, 0) = 1, C(n, k) = C(n-1, k-1) + C(n-1, k)$$

HW 2. 예시

Input	Output
5 0	1
10 7	120
12 6	924

HW 3. 이중 for문

Description

Given N ($1 \leq N \leq 1000$) and the scores of N students (integer, $0 \leq \text{score} \leq 10000$), print each student's rank.

The rank of the student A is defined as "the number of students who scored better than A " + 1.

HW 3. 예시

Input	Output
4 20 17 10 12	1 2 4 3
5 1 100 1 100 1	3 1 3 1 3

HW 4.

Description

There are N points on 2D plane. Find minimum and maximum "Manhattan distance" that you can obtain by choosing two distinct points.

Manhattan distance between two points $P1=(x1,y1)$, $P2=(x2,y2)$ is defined as

$$\text{dist} = |x1-x2| + |y1-y2|$$

HW 4.

Input

On the first line, number of points N is given. ($2 \leq N \leq 1000$).

On the following N lines, each points' x and y coordinates are given.

($|x|, |y| \leq 10000$, all coordinates are integer)

output

Print two lines. Minimum and maximum manhattan distances.

HW 4. 예시

Input	Output
4 0 0 0 1 1 1 1 0	1 2
2 -1 -1 -1 -1	0 0
5 -7 9 3 8 0 -2 -4 8 3 9	1 18

안내 사항

- 실습과제를 완료하지 못하신 분은 10/15 오후 11:59 까지 채점 사이트에 업로드 해주세요.
- 배열을 사용할 때에는 항상 **초기화**를 해주세요
- 디버깅을 할 때, printf를 사용해서 변수 값을 확인하는 방법을 사용하세요