

[1003]피보나치 함수

1003번: 피보나치 함수

다음 소스는 N번째 피보나치 수를 구하는 C++ 함수이다. int fibonacci(int n) { if (n == 0) { printf("0"); return 0; } else if (n == 1) { printf("1"); return 1; } else { return fibonacci(n-1) +

<https://www.acmicpc.net/problem/1003>

BAEKJOON
ONLINE JUDGE

문제

다음 소스는 N번째 피보나치 수를 구하는 C++ 함수이다.

```
1 int fibonacci(int n) {
2     if (n == 0) {
3         printf("0");
4         return 0;
5     } else if (n == 1) {
6         printf("1");
7         return 1;
8     } else {
9         return fibonacci(n-1) + fibonacci(n-2);
10    }
11 }
```

fibonacci(3) 을 호출하면 다음과 같은 일이 일어난다.

- fibonacci(3) 은 fibonacci(2) 와 fibonacci(1) (첫 번째 호출)을 호출한다.
- fibonacci(2) 는 fibonacci(1) (두 번째 호출)과 fibonacci(0) 을 호출한다.
- 두 번째 호출한 fibonacci(1) 은 1을 출력하고 1을 리턴한다.
- fibonacci(0) 은 0을 출력하고, 0을 리턴한다.
- fibonacci(2) 는 fibonacci(1) 과 fibonacci(0) 의 결과를 얻고, 1을 리턴한다.
- 첫 번째 호출한 fibonacci(1) 은 1을 출력하고, 1을 리턴한다.
- fibonacci(3) 은 fibonacci(2) 와 fibonacci(1) 의 결과를 얻고, 2를 리턴한다.

1은 2번 출력되고, 0은 1번 출력된다. N이 주어졌을 때, fibonacci(N) 을 호출했을 때, 0과 1이 각각 몇 번 출력되는지 구하는 프로그램을 작성하시오.

입력

첫째 줄에 테스트 케이스의 개수 T가 주어진다.

각 테스트 케이스는 한 줄로 이루어져 있고, N이 주어진다. N은 40보다 작거나 같은 자연수 또는 0이다.

출력

각 테스트 케이스마다 0이 출력되는 횟수와 1이 출력되는 횟수를 공백으로 구분해서 출력한다.

코드

```
package me.jihoho.baekjoon.dp.b1003;

import java.io.*;
import java.util.Scanner;

/**
 * Created by IntelliJ IDEA
 * User: hojun
 * Date: 2021-03-17 Time:오후3:27
 */
public class Main {
    public static void main(String[] args) throws IOException {
        Node[] cache=new Node[41];
        // 객체 배열 초기화
        for(int i=0;i<cache.length;i++){
            cache[i]=new Node();
        }
        int t;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bw=new BufferedWriter(new OutputStreamWriter(System.out));

        t=Integer.parseInt(br.readLine());
        int[] arr=new int[t];
        for(int i=0;i<t;i++){
            arr[i]=Integer.parseInt(br.readLine());
        }

        for(int i=0;i<arr.length;i++){
            Node result=recursive(arr[i], cache);
            bw.write(result.zeroCnt+" "+ result.oneCnt+"\n");
        }

        br.close();
        bw.close();
    }

    public static Node recursive(int n, Node[] cache){
        if(cache[n].zeroCnt!=0 || cache[n].oneCnt!=0){
            return cache[n];
        }
        if(n==0){
            cache[n].zeroCnt=1;
        }
    }
}
```

```

        return cache[n];
    }else if (n==1){
        cache[n].oneCnt=1;
        return cache[n];
    }else{
        cache[n].zeroCnt=recursive(n-1, cache).zeroCnt+recursive(n-2, cache).zeroCnt
;
        cache[n].oneCnt=recursive(n-1, cache).oneCnt+recursive(n-2, cache).oneCnt;
        return cache[n];
    }
}

class Node{
    int zeroCnt=0;
    int oneCnt=0;
}

```

이슈

- **Scanner** 클래스를 통해 입력 받았을 경우에 시간 초과가 계속 뜸.
- 알고 보니 Scanner 속도가 매우 느림.(c++로 알고리즘 풀다 보니 인지하지 못함)
- BufferedReader를 이용해서 입력을 받아보니 해결됨.

입력 방식에 대한 실행 속도

언어	입력방식	수행시간(초)
C/C++	scanf	0.798
C/C++	getchar(*)	0.390
C++	std::cin	2.051
C++	std::ios::sync_with_stdio(false) + std::cin	0.796
java	java.util.Scanner	6.068
java	java.io.BufferedReader(*)	0.934
golang	fmt.Scan	44.557
golang	bufio.Reader + fmt.Fscan	9.899
golang	bufio.Scanner(*)	1.299

참고

<https://carpediem0212.tistory.com/11>