

Fake Coin



[문제] 똑같은 모양의 금화 21개가 있는데 이 중에서 **약간 가벼운 가짜 금화**가 1개 있음을 알고 있다. 우리는 양팔저울(balance scale)을 이용해서 이 가짜 금화 1개를 찾아내려고 한다. 단 이 작업에서 양팔 저울을 가능한 적게 사용해서 이 하나의 가짜 금화를 찾아내려고 한다. 여러분은 이 저울을 대신할 내장 함수 `int balance(int left[], int right[])` 만을 사용해서 그 가짜 금화를 특정해야 한다. 21개의 금화는 정수 $\{1, 2, \dots, 21\}$ 으로 표시된다.

두 개의 정수 배열(integer array) `int left[22], int right[22]`을 이용해서 양팔 저울의 양쪽, 오른쪽 왼쪽 접시에 올린 동전을 표시한다. 만일 i 번째 동전을 오른쪽(왼쪽)에 올릴 경우에는 `int right[i]=1(left[i]=1)`로 표시한다. 예를 들어 왼쪽, 오른쪽 저울에 각각 $\{3, 4, 9, 10\}$, $\{1, 12, 13, 15\}$ 를 올려서 측정을 한다면 두 배열은 다음과 같이 준비되어야 한다. 배열에서 빈 곳은 숫자 zero(0)가 저장되어 있다.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
left[22]				1	1					1	1											
right[22]		1											1	1		1						

이 측정 함수 `int balance()`의 다음과 같다. 만일 +1이면 왼쪽이 더 무거운 경우, -1이면 오른쪽이 더 무거운 경우, 만일 숫자 0이면 양쪽의 무게가 같아 평형을 이룬을 의미한다. 만일 동일한 동전을 양쪽에 올리거나 표시한 된 숫자가 $\{0, 1\}$ 이 아닌 경우는 프로그램은 함수 `int balance()`를 부르는 과정에서 바로 종료되고 1회 시행한 것으로 count된다. 즉 아래와 같다.

<code>int balance() == +1</code>	왼쪽(left)이 더 무거움
<code>int balance() == 0</code>	양쪽의 무게가 같음
<code>int balance() == -1</code>	오른쪽(right)이 더 무거움
<code>int balance() == 999</code>	기타 오류

여러분은 `int balance()`의 호출만을 이용해서 가짜 동전을 찾아야 한다. 찾아낸 가짜 동전이 i 인 경우에는 `void report_coin(i)`를 활용하여 보고해야 한다. 이 함수를 부르면 프로그램은 종료된다. 즉 이번 문제에서 입출력 파일은 없으면 아래 예와 같이 반드시 head library를 추가해야 한다. 그리고 만일 시스템에서 지정한 횟수 이상으로 `int balance()` 호출하게 되면 프로그램은 자동으로 종료된다.

시작 프로그램의 preprocessor 부분에 `fakecoin.h`를 include 시켜야 한다. 이 헤드파일에 전역변수로 `int left[22], right[22]`가 선언되어 있다. 그리고 시작할 때 `load_coins()`를 부르면 입력이 완료된다. 이후로는 `balance(int left[], int right[])`를 이용해서 가짜 동전을

찾아서 보고해야 한다.

```
#include "fakecoin.h"    // 반드시 포함

int main( ) {

    load_coins( ) ;      // 입력을 위해서 제일 먼저 호출

    .....
    report_coin(i);      // 이 함수를 부르면 종료됨
}
```

[입출력] source 프로그램의 시작부분 head에 "**fakecoin.h**"를 반드시 #include로 넣어야 한다. 이 문제에서 여러분이 직접 stdin이나 stdout 파일을 조작하는 일은 없다.

[제한조건] 이번 문제는 C/C++로만 가능하다. 이름은 **fakecoin.{c, cpp}**, 제출 횟수와 마감, 최대 token의 개수는 NESPA에서 확인할 수 있다. NESPA 문제 게시판에 "**fakecoin.h**"가 제공되므로 이것을 이용해서 자신의 답이 맞는지 확인해 볼 수 있다. 단 실제 평가에 사용되는 헤더파일과는 다르다.