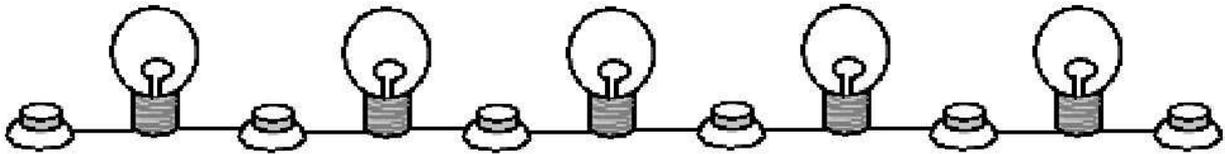


7. Bulb Game

문제 유형 : Interactive

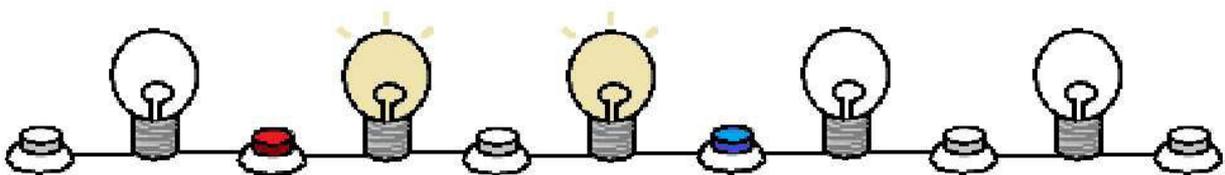
시간 제한 : 1 초, 메모리 제한 : 64 MB

친한 친구 민형이와 선율이가 재미있는 게임을 한다. 게임은 아래와 같이 N 개의 버튼과 $N-1$ 개의 전구가 있는 게임판에서 이루어진다. 여기서 N 은 항상 짝수이다.

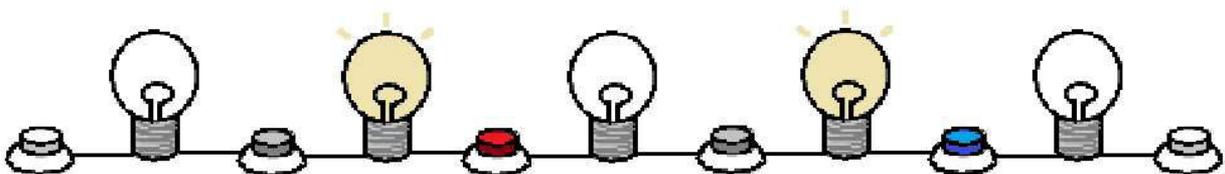


게임은 민형이와 선율이가 번갈아 버튼을 누르는 방식으로 진행된다. 매 순간마다, 민형이와 선율은 이때까지 눌러지 않은 버튼 중에서 하나를 골라 누른다. 선율이가 버튼을 누르면 민형이가 방금 전에 눌렀던 버튼과 선율이가 누른 버튼 사이에 있는 모든 전구의 상태가 바뀐다. (꺼져있는 전구는 켜지고, 켜있는 전구는 꺼진다.)

처음에는 모든 전구가 꺼져있다. 이 상태에서 민형이가 2번째 버튼을 누르고 선율이가 4번째 버튼을 누르면 게임판의 상태가 아래와 같이 바뀐다.



여기에서 민형이가 3번째 버튼을 누르고 선율이가 5번째 버튼을 누르면 게임판의 상태가 아래와 같이 바뀐다.



민형이와 선율이가 N 개의 버튼을 모두 누르면 게임이 끝난다. 게임이 끝난 후, 켜진 전구가 꺼진 전구보다 더 많으면 선율이가 이기고, 그 이외에는 민형이가 이긴다.

선올이는 게임의 긴장감을 조성하기 위해 진 사람이 민형, 선올 둘을 포함한 16명의 친구들에게 치킨을 쏘겠다는 제안을 했다. 알다시피 16명분의 치킨은 매우 비싸기 때문에 선올이가 게임에서 진다면 기껏 파격적인 제안을 했다가 거액을 내야 하는 대참사가 일어날 수 있다. 선올이가 이길 수 있도록 전략을 짜는 프로그램을 작성하여라. 물론 선올이가 이길 수 있는 전략은 존재한다.

요구사항

당신은 게임을 하는 함수 `Init(N)` 과 `MakeTurn(M)` 을 작성해야 한다. (`Init(N)`, `MakeTurn(M)` 함수는 `bulb.c` / `bulb.cpp`에 있다.)

`void Init(int N)` 함수는 맨 처음에 한 번 호출되며, 여기서 전처리 등을 할 수 있다.

- `N` : 전구의 수 ($2 \leq N \leq 1,000$, `N`은 짝수)

`int MakeTurn(int M)` 함수는 `Init` 함수가 호출된 후 `N/2`번 호출되며, 당신은 선올이가 눌러야 하는 버튼의 번호를 반환해야 한다.

- `M` : 민형이가 자신의 턴에 누른 버튼의 번호 ($1 \leq M \leq N$)

- 반환값 : 선올이가 자신의 턴에 눌러야 하는 버튼의 번호 ($1 \leq \text{반환값} \leq N$)

만약 게임이 정상적으로 끝나고 선올이가 졌다면 'Wrong[1]'을 출력한다. `MakeTurn(M)` 함수의 반환값이 범위를 벗어나거나 이미 눌렀던 버튼의 번호라면 'Wrong[2]'를 출력하고 프로그램이 즉시 끝난다.

예시 프로그램은 다음과 같다.

```
#include "bulb.h"

void Init(int N) {
    // do nothing
}

int MakeTurn(int M) {
    return (M+2)%7;
}
```

샘플 인터페이스

문제 페이지에서 샘플 코드를 다운로드받을 수 있다. 만약 Visual Studio나 Eclipse, Code::Blocks 와 같은 IDE 툴을 사용한다면 `bulb.cpp`, `bulb.h`, `grader.cpp` (또는 `bulb.c`, `bulb.h`, `grader.c`)를 한 프로젝트에 넣어서 컴파일하면 된다. 터미널에서 코드를 컴파일한다면 대회 페이지에 있는 컴파일 명령어를 이용하면 된다.

프로그램을 실행한 후 표준입력(stdin)으로 N을 입력받은 후 민형이가 누른 버튼의 번호를 추가로 입력받는다. 그러면 채점기는 선율이가 누른 버튼의 번호를 출력해준다. 게임이 끝난 후 샘플 채점기는 정답/오답 여부를 출력한다.

답안을 제출할 때에는 bulb.c 또는 bulb.cpp만 제출하면 된다.

입출력 예제

입력 (stdin)	함수 호출	출력 (stdout)
6	Init(6)	
2	MakeTurn(2)	4
3	MakeTurn(3)	5
6	MakeTurn(6)	1
		Correct

참고

메모리 접근, 시스템 호출 등의 비정상적인 방법을 사용하면 오답 처리될 수 있음에 유의하여야.

This Page is intentionally left blank.