

## 2. LR Springboard

문제 유형 : Interactive

시간 제한 : 1 초, 메모리 제한 : 64 MB

병찬이는 놀이공원의 ‘스프링 야드’ 어트랙션을 관리하는 관리자이다. ‘스프링 야드’ 어트랙션은 스프링 N개가 일렬로 늘어진 모습을 하고 있다. 이 어트랙션에서는 한 스프링을 밟으면 높이 떠올랐다가 바로 옆에 있는 스프링으로 떨어지는데, 이런 과정이 반복되면 보는 사람은 짜릿함을, 즐기는 사람은 스릴을 느낄 수 있다. 만약 사람이 가장 왼쪽 스프링에서 왼쪽으로 떠오르거나 가장 오른쪽 스프링에서 오른쪽으로 떠오르면 매트에 떨어지며 한 번의 차례가 끝난다.



여기에 있는 스프링은 매우 특이하게 설계되어 있다. 모든 스프링은 왼쪽 또는 오른쪽으로 살짝 기울어져 있는데, 스프링을 밟으면 그 스프링이 보고 있는 방향으로 높이 떠오른다. 또, 사람이 어떤 스프링을 밟아서 떠오르면 그 스프링은 알아서 방향을 바꾼다. 왼쪽을 보고 있던 스프링은 오른쪽으로, 오른쪽을 보고 있던 스프링은 왼쪽으로 방향이 바뀐다.

벌써 날이 어둑어둑해져서 병찬이는 어트랙션을 정리하려고 한다. 병찬이는 놀이공원의 방침에 따라 모든 스프링이 왼쪽을 향하게 해야 한다. 병찬이가 자기 손으로 직접 스프링의 방향을 바꾸면 안전사고의 위험이 있으므로 그는 파란색 공을 이용해서 스프링의 방향을 바꾸려고 한다.



병찬이가 높은 곳에서 공을 떨어뜨리면 공은 이리저리 튀다가 왼쪽 끝 매트나 오른쪽 끝 매트에 떨어진다. 아래 그림과 같이 병찬이가 3번째 스프링에 공을 떨어뜨린 후 첫 번째 스프링에 공을 떨어뜨리면 모든 스프링이 왼쪽을 보게 할 수 있다.



허나, 병찬이가 높은 곳에서 스프링을 보면 N개의 스프링이 어떤 방향을 향하고 있는지 제대로 보이지 않는다. 즉, 병찬이는 처음부터 끝까지 스프링의 상태를 알 수 없다. 그렇지만 병찬이가 파란색 공을 떨어뜨렸을 때 그 공이 왼쪽 끝에서 멈추는지 오른쪽 끝에서 멈추는지는 쉽게 분간이 가능하다. 병찬이를 도와 공을 적절히 떨어뜨려서 모든 스프링이 왼쪽을 보게 하는 프로그램을 작성하여라.

## 요구사항

당신은 스프링을 가지런히 정리하는 함수 `Reorder(N)` 을 작성해야 한다. (`Reorder(N)` 함수는 `springboard.c` / `springboard.cpp`에 있다.)

`void Reorder(int N)` 함수의 인수는 다음과 같다.

- `N` : 스프링의 수 ( $1 \leq N \leq 12,345$ )

당신은 `Reorder` 함수 안에서 `PutBall(K)`을 몇 번 호출할 수 있다.

`int PutBall(int K)` 는 왼쪽에서 `K`번째 스프링에 공을 떨어뜨리는 함수이다.

- `K` : 공을 놓는 위치 ( $1 \leq K \leq N$ )
- 반환값 : 공이 왼쪽으로 나가면 `-1`, 오른쪽으로 나가면 `1`을 반환한다.

`Reorder(N)` 함수가 끝나면 모든 스프링이 왼쪽을 보아야 한다. 만약 그렇지 않다면 채점 프로그램은 `'Wrong[1]'`을 출력한다. 함수를 호출할 때 인자의 범위를 벗어난다면 `'Wrong[2]'`를 출력하고 프로그램이 즉시 끝난다. 만약 `PutBall(K)`을 특정 횟수를 초과하여 호출하면 `'Wrong[3]'`을 출력하고 프로그램이 즉시 끝난다.

`PutBall(K)` 함수는 **최대 16번** 호출할 수 있다.

예시 프로그램은 다음과 같다. 주석은 문제와 같은 상황에서 함수의 반환값을 의미한다.

```
#include "springboard.h"

void Reorder(int N) {
    PutBall(3); // 반환값 = -1
    PutBall(1); // 반환값 = 1
}
```

## 샘플 인터페이스

문제 페이지에서 샘플 코드를 다운로드받을 수 있다. 만약 Visual Studio나 Eclipse, Code::Blocks 와 같은 IDE 툴을 사용한다면 `springboard.cpp`, `springboard.h`, `grader.cpp` (또는 `springboard.c`, `springboard.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 `N`과 `N`개 스프링의 방향을 입력받는다. 스프링의 방향으로 들어오는 문자열은 `'L'` 또는 `'R'`로 이루어진 길이 `N`인 문자열이며, `'L'`은 왼쪽을 보고 있는 스프링을, `'R'`은 오른쪽을 보는 스프링을 의미한다. 모든 정보를 입력받으면 샘플 채점기는 정답/오답 여부를

출력한다.

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

### 입출력 예제

입력 (stdin)	출력 (stdout)
5 LLRRL	Correct

### 참고

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