

가벼운 돌과 무거운 돌 찾기 (stones)

당신 앞에 N 개의 돌이 놓여 있고, 당신은 이 중 가장 가벼운 돌과 무거운 돌이 무엇인지 알아내고자 한다. 정확한 무게를 측정할 수 있는 저울이 있다면 좋았겠으나, 당신이 가지고 있는 것은 양팔 저울 하나뿐이다.

당신이 할 수 있는 일은, 양팔 저울의 왼쪽 접시에 돌 하나를 올리고 오른쪽 접시에 다른 돌 하나를 올린 후, 양팔 저울이 어느 쪽으로 기울는지 보는 것이다. 만약 두 돌의 무게가 다르다면 양팔 저울은 무거운 쪽으로 기울 것이고, 무게가 같다면 양팔 저울은 평형을 이룰 것이다. 즉, 얻을 수 있는 결과는, '왼쪽 돌이 더 무겁다', '오른쪽 돌이 더 무겁다', '두 돌의 무게가 같다' 중 하나이다.

당신은 양팔 저울을 **900번 이하**만 사용하여 가장 가벼운 돌과 가장 무거운 돌을 모두 알아내고자 한다. 즉, 가장 가벼운 돌 또는 가장 무거운 돌이 여러 개 있을 수도 있고, 당신은 그 돌들을 모두 찾아야 한다. 모든 돌의 무게가 같다면, 모든 돌은 가장 무거우면서 가장 가볍다.

입력 형식

첫 번째 줄에 테스트 케이스의 수 T ($1 \leq T \leq 100$)가 주어진다. 이후에는 다음 과정을 T 번 반복해야 한다.

상호 작용(Interaction)

먼저 한 개의 줄에 돌의 개수 N ($2 \leq N \leq 600$)이 주어진다.

양팔 저울을 사용하려면, '? i j '를 한 개의 줄에 출력해야 한다. 여기서 i 는 왼쪽 접시에 올려질 돌의 번호이고, j 는 오른쪽 접시에 올려질 돌의 번호이다. i 와 j 는 $1 \leq i, j \leq N$, $i \neq j$ 를 모두 만족하는 정수여야 하며, 그렇지 않을 시 오답 판정을 받게 된다. 만약 i 번 돌이 j 번 돌보다 무겁다면 '>'가, 가볍다면 '<'가, 두 돌의 무게가 같다면 '='가 한 개의 줄에 주어진다. 양팔 저울은 한 테스트 케이스마다 900번까지 사용할 수 있고, 901번 이상 사용한다면 오답 판정을 받게 된다.

가장 가벼운 돌과 가장 무거운 돌을 모두 알아냈다면, 우선 '!'만 있는 한 개의 줄을 출력해야 한다. 그 다음 한 개의 줄에는, 가장 가벼운 돌의 개수 n_1 을 먼저 출력하고, 공백 하나를 출력한 뒤, 가장 가벼운 돌(들)의 번호 p_1, p_2, \dots, p_{n_1} 를 공백 하나씩을 사이로 두고 출력한다. 그 다음 한 개의 줄에는, 가장 무거운 돌의 개수 n_2 를 먼저 출력하고, 공백 하나를 출력한 뒤, 가장 무거운 돌(들)의 번호 q_1, q_2, \dots, q_{n_2} 를 공백 하나씩을 사이로 두고 출력한다. 같은 번호를 여러 번 출력하면 안 되지만, 돌의 번호를 출력하는 순서는 관계 없다. 즉, 정답 또는 오답 여부는 실제로 가벼운 돌과 무거운 돌을 정확히 찾았는지에 의해서만 결정된다.

매 줄을 출력할 때마다 C/C++에서는 'fflush(stdout);', C++에서는 'std::cout << std::flush;', Java에서는 'System.out.flush();'를 통해 표준 출력 버퍼를 비워줘야 한다.

여러분의 구현을 돕기 위해, 채점 시스템에서 C/C++/Java 예시 코드를 내려받을 수 있다. 이미 구현되어 있는 'weigh' 함수와 'report' 함수를 그대로 사용하면 된다.

채점 방식

- (1점) $N \leq 30$
- (8점) $N \leq 42$
- (10점) $N \leq 225$
- (25점) $N \leq 450$
- (35점) 모든 돌의 무게는 서로 다르다.
- (21점) 추가 제약 조건 없음.

참고

문제 페이지 하단에서 내려받을 수 있는 `helper.zip`에는 아래 파일들이 들어 있다.

- `stones.(c|cpp|java)`: 예시 코드이다. “/* IMPLEMENT HERE */” 사이의 부분을 구현하면 된다. 양팔 저울을 $N(N-1)$ 번 사용하는 알고리즘이 구현되어 있다. 이 코드를 그대로 제출하면 1점을 받을 수 있다.
- `simulator.py`: 실제 채점 시스템처럼 프로그램과 입출력을 주고 받는 Python 스크립트이다. Windows 10, Ubuntu 18.04 (WSL), Mac OS X의 Python 2.7, 3.6에서 테스트되었으며, Python 버전이 다르면 실행이 제대로 되지 않을 수 있다.

– 간단한 사용법: 명령줄에 ‘`python simulator.py -h`’를 입력하면 사용 가능한 옵션들을 볼 수 있다.

* `--input INPUT`: 이 옵션을 사용하면 스크립트가 `INPUT` 파일에서 테스트 데이터를 읽는다. 입력 형식은 아래 ‘`example.in`’ 부분을 참고한다.

* `-l, --log`: 이 옵션을 켜면, 스크립트와 여러분이 작성한 프로그램 사이의 입출력 경과를 모두 출력한다.

* `--limit LIMIT`: 양팔 저울을 사용하는 횟수 제한이다. 명시하지 않았을 때의 기본값은 문제에 제시한 것과 같은 900이다.

* 모든 옵션을 입력한 뒤, 마지막에 코드를 실행하는 명령어를 입력해야 한다. 명령어를 따옴표로 감싸면 안 된다.

– 예시:

* `python simulator.py --limit 100 -l ./stones`

* `python simulator.py --log stones.exe`

* `python simulator.py --input input.txt java stones`

- `example.in`: 아래의 형식으로 돌의 무게에 대한 정보를 스크립트에게 제공한다.

– 1번째 줄: T

– 다음을 T 번 반복:

* 1번째 줄: N

* 2번째 줄: $w_1 \ w_2 \ \dots \ w_N$ (여기서, w_i 는 i 번 돌의 무게이며, 정수여야 한다.)