

분실 카드



[문제] 1부터 N 까지의 정수가 쓰여진 N 장의 카드로 게임을 준비하고 있다. 이 N 장의 카드는 특별하게 고안된 셔플 장치(Card Shuffler)¹⁾를 이용해서 충분히 랜덤하게 섞는다. 그런데 셔플 장치에 카드를 넣는 과정에서 몇 장의 카드가 분실되었다는 사실을 알게 되었다. 이 분실된 카드는 최대 2장이라는 사실도 알고 있다. 여러분은 이 셔플 장치에서 나오는 카드를 모두 읽은 뒤 그 분실된 카드 번호를 찾아야 한다.

[입출력] 입력과 출력은 모든 표준 입출력을 사용한다. 입력파일 `stdin`의 첫 줄에 정수 N 이 주어진다. 단 $5 \leq N \leq 10,000$ 이다. 이어지는 $N-1$ 혹은 $N-2$ 개의 각 줄에는 셔플 장치에서 뽑힌 카드 번호가 한 줄에 하나씩 제시된다. 여러분은 이것을 모두 읽은 다음 분실된 카드 번호를 찾아서 한 줄에 하나씩, 최대 2개의 줄에 각각 순서대로 출력해야한다. 단 빠진 카드가 2개일 경우에는 작은 번호부터 먼저 출력해야 한다.

[예제]

stdin	stdout	stdin	stdout
10 // N	3	100 // N	56
8	6	98	
4		41	
1		... (중략)	
10		...	
5		31	
9		7	
2		65	
7 // 8번째		49 // 99번째	

[제한조건] 제출 프로그램은 `missing.{c, cpp, java}`이다. 이번 문제는 python code를 허용하지 않는다. 제출횟수는 최대 15회, 각 데이터 당 수행 제한시간은 1초이다. 과제 마감시간은 **3월 8일(월) 저녁 10시이며**, 제출은 2월 27일부터 가능하다. 단 여러분의 코드에서 사용할 수 있는 token의 최대 갯수는 **200**이다. token은 예약어(reserved words), 연산자, 숫자, 변수 등으로 구성되며 comment 문장은 token 계산에 포함되지 않는다. NESPA page의 상단 tab 메뉴를 이용하면 여러분의 코드에 포함된 token 수를 확인할 수 있다.

[제한조건] 크기 N 인 배열 `int numbers[N]`를 이용하면 쉽게 해결할 수 있지만 이 경우에는 크기가 확정되지 않은 정수 배열을 준비해서 저장하고, 그 내용을 다시 검색해야 하는 부담이 있다. 여러분은 이보다 더 간략하고 우아한 알고리즘을 찾아야 한다.

1) 카지노(casino)에서는 위 그림과 같이 셔플링(shuffling) 장치를 이용하여 게임의 공정성과 딜러의 부정을 방지한다.