1. 정수를 저장하는 스택을 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 다섯 가지이다.

- push X: 정수 X를 스택에 넣는 연산이다.
- pop: 스택에서 가장 위에 있는 정수를 빼고, 그 수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- size: 스택에 들어있는 정수의 개수를 출력한다.
- empty: 스택이 비어있으면 1, 아니면 0을 출력한다.
- top: 스택의 가장 위에 있는 정수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.

예제)

입력	출력
14	
push 1	
push 2	2
top	2
size	0
empty	2
pop	1
pop	-1
pop	0
size	1
empty	-1
pop	0
push 3	3
empty	
top	

2. Your boss has asked you to add up a sequence of positive numbers to determine how much money your company made last year.

Unfortunately, your boss reads out numbers incorrectly from time to time.

Fortunately, your boss realizes when an incorrect number is read and says “zero”, meaning “ignore the current last number.”

Unfortunately, your boss can make repeated mistakes, and says “zero” for each mistake.

For example, your boss may say “One, three, five, four, zero, zero, seven, zero, zero, six”, which means the total is 7 as explained in the following chart:

Boss statement(s)	Current numbers	Explanation
“One, three, five, four”	1, 3, 5, 4	Record the first four numbers.
“zero, zero“	1, 3	Ignore the last two numbers.
“seven”	1, 3, 7	Record the number 7 at the end of our list.
“zero, zero”	1	Ignore the last two numbers.
“six”	1, 6	We have read all numbers, and the total is 7.

At any point, your boss will have said at least as many positive numbers as “zero” statements. If all positive numbers have been ignored, the sum is zero.

Write a program that reads the sequence of boss statements and computes the correct sum.

▶ Input

The first line of input contains the integer K ($1 \leq K \leq 100\ 000$) which is the number of integers (including “zero”) your boss will say. On each of the next K lines, there will either be one integer between 1 and 100 (inclusive), or the integer 0.

▶ Output

The output is one line, containing the integer which is the correct sum of the integers read, taking the “zero” statements into consideration. You can assume that the output will be an integer in the range 0 and 1 000 000 (inclusive).

Example)

Input	Output
4 3 0 4 0	0

3. 스택 (stack)은 기본적인 자료구조 중 하나로, 컴퓨터 프로그램을 작성할 때 자주 이용되는 개념이다. 스택은 자료를 넣는 (push) 입구와 자료를 뽑는 (pop) 입구가 같아 제일 나중에 들어간 자료가 제일 먼저 나오는 (LIFO, Last in First out) 특성을 가지고 있다.

1부터 n까지의 수를 스택에 넣었다가 뽑아 늘어놓음으로써, 하나의 수열을 만들 수 있다. 이때, 스택에 push하는 순서는 반드시 오름차순을 지키도록 한다고 하자. 임의의 수열이 주어졌을 때 스택을 이용해 그 수열을 만들 수 있는지 없는지, 있다면 어떤 순서로 push와 pop 연산을 수행해야 하는지를 알아낼 수 있다. 이를 계산하는 프로그램을 작성하라.

입력
첫 줄에 n (1 ≤ n ≤ 100,000)이 주어진다. 둘째 줄부터 n개의 줄에는 수열을 이루는 1이상 n이하의 정수가 하나씩 순서대로 주어진다. 물론 같은 정수가 두 번 나오는 일은 없다.

출력
입력된 수열을 만들기 위해 필요한 연산을 한 줄에 한 개씩 출력한다. push연산은 +로, pop 연산은 -로 표현하도록 한다. 불가능한 경우 NO를 출력한다.

예제)

입력	출력
8 4 3 6 8 7 5 2 1	+ + + + - - + + - + + - - - - -

4. N장의 카드가 있다. 각각의 카드는 차례로 1부터 N까지의 번호가 붙어 있으며, 1번 카드가 제일 위에, N번 카드가 제일 아래인 상태로 순서대로 카드가 놓여 있다.

이제 다음과 같은 동작을 카드가 한 장 남을 때까지 반복하게 된다. 우선, 제일 위에 있는 카드를 바닥에 버린다. 그 다음, 제일 위에 있는 카드를 제일 아래에 있는 카드 밑으로 옮긴다.

예를 들어 N=4인 경우를 생각해 보자. 카드는 제일 위에서부터 1234 의 순서로 놓여있다. 1을 버리면 234가 남는다. 여기서 2를 제일 아래로 옮기면 342가 된다. 3을 버리면 42가 되고, 4를 밑으로 옮기면 24가 된다. 마지막으로 2를 버리고 나면, 남는 카드는 4가 된다.

N이 주어졌을 때, 제일 마지막에 남게 되는 카드를 구하는 프로그램을 작성하시오.

- 입력
첫째 줄에 정수 N(1 ≤ N ≤ 500,000)이 주어진다.

- 출력
첫째 줄에 남게 되는 카드의 번호를 출력한다.

예제)

입력	출력
6	4

5. 정수를 저장하는 덱(Deque)를 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 여덟 가지이다.

- push_front X: 정수 X를 덱의 앞에 넣는다.
- push_back X: 정수 X를 덱의 뒤에 넣는다.
- pop_front: 덱의 가장 앞에 있는 수를 빼고, 그 수를 출력한다. 만약, 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- pop_back: 덱의 가장 뒤에 있는 수를 빼고, 그 수를 출력한다. 만약, 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- size: 덱에 들어있는 정수의 개수를 출력한다.
- empty: 덱이 비어있으면 1을, 아니면 0을 출력한다.
- front: 덱의 가장 앞에 있는 정수를 출력한다. 만약 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- back: 덱의 가장 뒤에 있는 정수를 출력한다. 만약 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.

입력
첫째 줄에 주어지는 명령의 수 N (1 ≤ N ≤ 10,000)이 주어진다. 둘째 줄부터 N개의 줄에는 명령이 하나씩 주어진다. 주어지는 정수는 1보다 크거나 같고, 100,000보다 작거나 같다. 문제에 나와있지 않은 명령이 주어지는 경우는 없다.

출력
출력해야하는 명령이 주어질 때마다, 한 줄에 하나씩 출력한다.

예제)

입력	출력
15	2
push_back 1	1
push_front 2	2
front	0
back	2
size	1
empty	-1
pop_front	0
pop_back	1
pop_front	-1
size	0
empty	3

pop_back push_front 3 empty front	