스택

## **문제**

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.

## **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 1 복사**

14

push 1

push 2

top

size

empty

pop

pop

pop

size

empty

pop

push 3

empty

top

## **예제 출력 1 복사**

2

2

0

2

1

-1

0

1

-1

0

3

## **예제 입력 2 복사**

7

pop

top

push 123

top

pop

top

pop

## **예제 출력 2 복사**

-1

-1

123

123

-1

-1

제로

## **문제**

나코더 기장 재민이는 동아리 회식을 준비하기 위해서 장부를 관리하는 중이다.

재현이는 재민이를 도와서 돈을 관리하는 중인데, 애석하게도 항상 정신없는 재현이는 돈을 실수로 잘못 부르는 사고를 치기 일쑤였다.

재현이는 잘못된 수를 부를 때마다 0을 외쳐서, 가장 최근에 재민이가 쓴 수를 지우게 시킨다.

재민이는 이렇게 모든 수를 받아 적은 후 그 수의 합을 알고 싶어 한다. 재민이를 도와주자!

## **입력**

첫 번째 줄에 정수 K가 주어진다. (1 ≤ K ≤ 100,000)

이후 K개의 줄에 정수가 1개씩 주어진다. 정수는 0에서 1,000,000 사이의 값을 가지며, 정수가 "0" 일 경우에는 가장 최근에 쓴 수를 지우고, 아닐 경우 해당 수를 쓴다.

정수가 "0"일 경우에 지울 수 있는 수가 있음을 보장할 수 있다.

## **출력**

재민이가 최종적으로 적어 낸 수의 합을 출력한다. 최종적으로 적어낸 수의 합은 231-1보다 작거나 같은 정수이다.

## **예제 입력 1 복사**

4

3

0

4

0

## **예제 출력 1 복사**

0

## **예제 입력 2 복사**

10

1

3

5

4

0

0

7

0

0

6

## **예제 출력 2 복사**

7

괄호

## **문제**

괄호 문자열(Parenthesis String, PS)은 두 개의 괄호 기호인 ‘(’ 와 ‘)’ 만으로 구성되어 있는 문자열이다. 그 중에서 괄호의 모양이 바르게 구성된 문자열을 올바른 괄호 문자열(Valid PS, VPS)이라고 부른다. 한 쌍의 괄호 기호로 된 “( )” 문자열은 기본 VPS 이라고 부른다. 만일 x 가 VPS 라면 이것을 하나의 괄호에 넣은 새로운 문자열 “(x)”도 VPS 가 된다. 그리고 두 VPS x 와 y를 접합(concatenation)시킨 새로운 문자열 xy도 VPS 가 된다. 예를 들어 “(())()”와 “((()))” 는 VPS 이지만 “(()(”, “(())()))” , 그리고 “(()” 는 모두 VPS 가 아닌 문자열이다.

여러분은 입력으로 주어진 괄호 문자열이 VPS 인지 아닌지를 판단해서 그 결과를 YES 와 NO 로 나타내어야 한다.

## **입력**

입력 데이터는 표준 입력을 사용한다. 입력은 T개의 테스트 데이터로 주어진다. 입력의 첫 번째 줄에는 입력 데이터의 수를 나타내는 정수 T가 주어진다. 각 테스트 데이터의 첫째 줄에는 괄호 문자열이 한 줄에 주어진다. 하나의 괄호 문자열의 길이는 2 이상 50 이하이다.

## **출력**

출력은 표준 출력을 사용한다. 만일 입력 괄호 문자열이 올바른 괄호 문자열(VPS)이면 “YES”, 아니면 “NO”를 한 줄에 하나씩 차례대로 출력해야 한다.

## **예제 입력 1 복사**

6

(())())

(((()())()

(()())((()))

((()()(()))(((())))()

()()()()(()()())()

(()((())()(

## **예제 출력 1 복사**

NO

NO

YES

NO

YES

NO

## **예제 입력 2 복사**

3

((

))

())(()

## **예제 출력 2 복사**

NO

NO

NO

스택수열

## **문제**

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 1 복사**

8

4

3

6

8

7

5

2

1

## **예제 출력 1 복사**

+

+

+

+

-

-

+

+

-

+

+

-

-

-

-

-

## **예제 입력 2 복사**

5

1

2

5

3

4

## **예제 출력 2 복사**

NO

균형잡힌 세상

## **문제**

세계는 균형이 잘 잡혀있어야 한다. 양과 음, 빛과 어둠 그리고 왼쪽 괄호와 오른쪽 괄호처럼 말이다.

정민이의 임무는 어떤 문자열이 주어졌을 때, 괄호들의 균형이 잘 맞춰져 있는지 판단하는 프로그램을 짜는 것이다.

문자열에 포함되는 괄호는 소괄호("()") 와 대괄호("[]")로 2종류이고, 문자열이 균형을 이루는 조건은 아래와 같다.

* 모든 왼쪽 소괄호("(")는 오른쪽 소괄호(")")와만 짝을 이뤄야 한다.
* 모든 왼쪽 대괄호("[")는 오른쪽 대괄호("]")와만 짝을 이뤄야 한다.
* 모든 오른쪽 괄호들은 자신과 짝을 이룰 수 있는 왼쪽 괄호가 존재한다.
* 모든 괄호들의 짝은 1:1 매칭만 가능하다. 즉, 괄호 하나가 둘 이상의 괄호와 짝지어지지 않는다.
* 짝을 이루는 두 괄호가 있을 때, 그 사이에 있는 문자열도 균형이 잡혀야 한다.

정민이를 도와 문자열이 주어졌을 때 균형잡힌 문자열인지 아닌지를 판단해보자.

## **입력**

하나 또는 여러줄에 걸쳐서 문자열이 주어진다. 각 문자열은 영문 알파벳, 공백, 소괄호("( )") 대괄호("[ ]")등으로 이루어져 있으며, 길이는 100글자보다 작거나 같다.

입력의 종료조건으로 맨 마지막에 점 하나(".")가 들어온다.

## **출력**

각 줄마다 해당 문자열이 균형을 이루고 있으면 "yes"를, 아니면 "no"를 출력한다.

## **예제 입력 1 복사**

So when I die (the [first] I will see in (heaven) is a score list).

[ first in ] ( first out ).

Half Moon tonight (At least it is better than no Moon at all].

A rope may form )( a trail in a maze.

Help( I[m being held prisoner in a fortune cookie factory)].

([ (([( [ ] ) ( ) (( ))] )) ]).

.

.

## **예제 출력 1 복사**

yes

yes

no

no

no

yes

yes

오큰수

## **문제**

크기가 N인 수열 A = A1, A2, ..., AN이 있다. 수열의 각 원소 Ai에 대해서 오큰수 NGE(i)를 구하려고 한다. Ai의 오큰수는 오른쪽에 있으면서 Ai보다 큰 수 중에서 가장 왼쪽에 있는 수를 의미한다. 그러한 수가 없는 경우에 오큰수는 -1이다.

예를 들어, A = [3, 5, 2, 7]인 경우 NGE(1) = 5, NGE(2) = 7, NGE(3) = 7, NGE(4) = -1이다. A = [9, 5, 4, 8]인 경우에는 NGE(1) = -1, NGE(2) = 8, NGE(3) = 8, NGE(4) = -1이다.

## **입력**

첫째 줄에 수열 A의 크기 N (1 ≤ N ≤ 1,000,000)이 주어진다. 둘째에 수열 A의 원소 A1, A2, ..., AN (1 ≤ Ai ≤ 1,000,000)이 주어진다.

## **출력**

총 N개의 수 NGE(1), NGE(2), ..., NGE(N)을 공백으로 구분해 출력한다.

## **예제 입력 1 복사**

4

3 5 2 7

## **예제 출력 1 복사**

5 7 7 -1

## **예제 입력 2 복사**

4

9 5 4 8

## **예제 출력 2 복사**

-1 8 8 -1