

1. 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	출력
So when I die (the [first] I will see in (heaven) is a score list).	yes
[ first in ] ( first out ).	yes
Half Moon tonight (At least it is better than no Moon at all).	no
A rope may form )( a trail in a maze.	no
Help( I[m being held prisoner in a fortune cookie factory)).	no
([ (([ [ ] ) ( ) (( ))) ) ).	yes
.	yes
.	