문제 D. Regulating Bracket Sequence

시간 제한 1 초 메모리 제한 1024 MB

다음 조건을 만족하는 문자열을 **균형 잡힌 괄호 문자열**이라고 한다.

- 빈 문자열은 균형 잡힌 괄호 문자열이다.
- S가 균형 잡힌 괄호 문자열이라면 (S), $\{S\}$, $[S]$도 각각 균형 잡힌 괄호 문자열이다.
- S와 T가 둘 다 균형 잡힌 괄호 문자열이라면 ST도 균형 잡힌 괄호 문자열이다.
- 위 방식으로 만들 수 없는 문자열은 균형 잡힌 괄호 문자열이 아니다.

두 균형 잡힌 괄호 문자열 S와 T가 주어질 때, 다음 연산을 0회 이상 사용하여 S를 T로 만들 수 있는지 판별해보자.

- 두 균형 잡힌 괄호 문자열 A와 B에 대해 S가 $(A)B$라면 S를 $(B)A$ 또는 $B(A)$로 바꾼다.
- 두 균형 잡힌 괄호 문자열 A와 B에 대해 S가 $\{A\}B$라면 S를 $\{B\}A$ 또는 $B\{A\}$로 바꾼다.
- 두 균형 잡힌 괄호 문자열 A와 B에 대해 S가 $[A]B$라면 S를 $[B]A$ 또는 $B[A]$로 바꾼다.

입력

첫 번째 줄에 균형 잡힌 괄호 문자열 S가 주어진다. ($2 \le |S| \le 500\,000$)

두 번째 줄에 균형 잡힌 괄호 문자열 T가 주어진다. ($2 \le |T| \le 500\,000$)

출력

주어진 연산을 0회 이상 사용하여 S를 T로 만들 수 있다면 **YES**를, 아니면 **NO**를 출력하여라.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
({}[]) [](){}	YES
()[]{} {}[]()	NO

첫 번째 예제의 경우 연산을 다음과 같이 적용하여 S를 T로 만들 수 있다.

- ({}[])에 연산을 적용하여 (){}[]로 만든다.
- (){}[]에 연산을 적용하여 {}[](로 만든다.
- {}[]()에 연산을 적용하여 []({}로 만든다.

두 번째 예제의 경우 연산을 어떻게 적용해도 S를 T로 만들 수 없다.