A string S consisting of N characters is called *properly nested* if:

* S is empty;
* S has the form "(U)" where U is a properly nested string;
* S has the form "VW" where V and W are properly nested strings.

For example, string "(()(())())" is properly nested but string "())" isn't.

Write a function:

class Solution { public int solution(string S); }

that, given a string S consisting of N characters, returns 1 if string S is properly nested and 0 otherwise.

For example, given S = "(()(())())", the function should return 1 and given S = "())", the function should return 0, as explained above.

Assume that:

* N is an integer within the range [0..1,000,000];
* string S consists only of the characters "(" and/or ")".

Complexity:

* expected worst-case time complexity is O(N);
* expected worst-case space complexity is O(1) (not counting the storage required for input arguments).