

January 2023 CSE 106

Online: Stack

Time: 25 minutes

Subsections C1 & C2

A string is a valid parentheses string (denoted **VPS**) if it meets one of the following:

- It is an empty string "", or a single character not equal to "(" or ")",
- It can be written as AB (A concatenated with B), where A and B are **VPS**'s, or
- It can be written as (A), where A is a **VPS**.

We can similarly define the **nesting depth** $depth(S)$ of any VPS S as follows:

- $depth("") = 0$
- $depth(C) = 0$, where C is a string with a single character not equal to "(" or ")"
- $depth(A + B) = \max(depth(A), depth(B))$, where A and B are **VPS**'s.
- $depth("(" + A + ")") = 1 + depth(A)$, where A is a **VPS**.

For example, "", "()()", and "()()()" are **VPS**'s (with nesting depths 0, 1, and 2), and ")(" and "(()" are not **VPS**'s.

Given a **VPS** represented as string s , **return the nesting depth** of s .

Input

Take a **VPS** s as input.

Output

Print the nesting depth of s .

See the Sample I/O for further clarification.

Sample I/O

Input	Output
$(1+(2*3)+((8)/4))+1$	3 [Digit 8 is inside of 3 nested parentheses in the string]
$(1)+((2))+(((3)))$	3
$1+(2*3)/(2-1)$	1

Marks Distribution

Approach	Marks
Print the nesting depth	70%
Implement using Stack	100%

Please note that any usage of the internet is strictly prohibited during the assignment. Usage of any unfair means will be duly punished.