Problem 3 – Longest Odd-Even Sequence

You are given **n numbers** in format **(a_1) (a_2) … (a_n)**. Write a program to **find the longest odd-even alternating sub-sequence** inside the input sequence. Such subsequence starts at some position with odd number and continues with even number, then again odd number, etc. The opposite is also allowed: start with even number, then odd number, then even number, etc. The special **number 0 (zero)** is considered odd and even in the same time. For example, if we have the input sequence **(3) (22) (-18) (55) (44) (3) (21)**, the longest odd-even alternating subsequence has length 4 and it is: **(-18) (55) (44) (3)**.

Input

The input comes from the console. It consists of a **single line holding the input sequence of numbers**. All numbers are in brackets. Spaces can be put anywhere between the numbers, even at the sequence start and at the sequence end. The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

Print **the length of the longest alternating sub-sequence** at the console.

Constraints

- The input **numbers** will be integers in the range [-100…100].
- The count of the input numbers will be in the range [1…1000].
- Time limit: 0.3 sec. Memory limit: 16 MB.

Examples

Input	Output
`(3) (22) (-18) (55) (44) (3) (21)`	4
`(1)(2)(3)(4)(5)(6)(7)(8)(9)(10)`	10
`  ( 2 )  ( 33 ) (1) (4)   (  -1  )`	3
`(102)(103)(0)(105)  (107)(1)`	4
`(2) (2) (2) (2) (2)`	1