Tweedledee's Split Brackets

Time limit: *2000 ms*
Memory limit: *256 MB*

Tweedledee has a bracket sequence S with round and square brackets.

He begins an exercise with two empty sequences S_1 and S_2. For each bracket in S, from left to right, Tweedledee decides to append that bracket to either S_1 or S_2. He wants S_1 and S_2 to have equal length and both be valid bracket sequences. How can he split S into S_1 and S_2?

A valid bracket sequence is defined as follows:

- `()` and `[]` are valid bracket sequences;
- If S is a valid bracket sequence, then `( S )` and `[ S ]` are also valid bracket sequences;
- If Q and R are two valid bracket sequences, then the concatenation of them, $S = QR$, is also a valid bracket sequence.

Standard input

The input has a single line consisting of the bracket sequence S.

Standard output

For each bracket in s from left to right, output 1 to append the bracket to S_1, or 2 to append the bracket to S_2. Separate the numbers by single spaces. If there are multiple answers, output the lexicographically smallest sequence. If there is no solution to make the split, output the word `impossible`.

Constraints and notes

- $4 \le |S| \le 36$

Input	Output	Explanation
`([]())[]`	`1 1 1 2 1 2 2 2`	This answer splits the sequence into `([])` and `()[]`. Although `1 2 2 1 1 1 2 2` is also a valid split, `1 1 1 2 1 2 2 2` is lexicographically smaller.
`([)]`	`1 2 1 2`	The only valid answer is to split into one pair of round brackets, and one pair of square brackets.
`((]]`	`impossible`	
`()[])`	`impossible`	Although it is possible to split the sequence into two valid bracket sequences, their lengths cannot equal.
`([[()][())]])`	`1 1 1 2 1 1 2 2 1 2 2 2`	