

Brackets

Ivancho, just like you, wants to take part in competitions which makes him stand up to some not too complicated tasks. But he's new to programming, so he seeks help from you, good programmers. The task which is at hand is a slight modification of the brackets task.

For those of you who haven't seen a similar task before, there is a given string including all four kinds of brackets – ([{ < > }]). The goal is to check if the bracket syntax is right – i.e. if each opening bracket has a matching closing one and vice versa. This – ([]) – has a right syntax, but this – ([)] – does not, neither this – ([) or this – [}). The brackets doesn't have priority – for example there can be curly brackets inside square or normal brackets.

The modification of the task, that Ivancho seeks help for, includes random lowercase latin letters in between the brackets.

Write a program called **brackets** which tells if a given string has a right bracket syntax.

Input: In the input file **brackets.in** there is a string including only brackets and lowercase latin letters. There are no white spaces.

Output: The output file **brackets.out** should contain either 1 for right syntax or 0 for a wrong one.

Limits: The string contains no more than 3000 characters.

Time limit: 0.5 sec

Memory limit: 256 MB

Preliminary tests: 4

Final tests: 10

Sample tests:

brackets.in	brackets.out
(ag)(aa(jh[ku<k(kuk)mm{!}>pp]b)fgh)	1
(ag)(aa(jh[ku<k(kukmm{!}>pp]b)fgh)	0
br(ac{ke}t[s])	1
k(	0
]hfp}nanac	0