Brackets

Given a string comprising just the characters (,),{,},[,] determine if it is well-formed or not by applying the following rules:

- Each type of bracket needs to be first opened then closed
  - Good: () or [] or {}
  - Bad: (() or {}}
- You can only close the last bracket that was opened
  - Good: ({})
  - Bad: ({)}
- Inside parenthesis () only braces {} are allowed
  - Good: ({})
  - Bad: ([]) or (())
- Inside braces {} only square brackets [] are allowed
  - Good: {[]}
  - Bad: {()} or {{}}
- Any bracket can appear (directly) inside square brackets []
  - Good: [()] or [{}] or [[]] or [[[]]]
  - Bad: [([])]
- You can use a list of braces where a single one is allowed of that type
  - Good: [()()] or {[][()()]} or ()()
- An empty string is not valid a expression
- Any other characters than (){}[] will invalidate the string

For a given string print out *true* if the string is well-formed or *false* if otherwise.

Write a program that can read from *stdin*. Process all lines and print out the result to *stdout*.

Enhance the program so it uses a multi-threaded approach. Dispatch the actual strings to test to as many threads as the machine has cores. Because the order is not necessarily the same as the input provide the output with the referring index of the nth input like this:

```
0:true
1:false
```