Question 6

Complete the implementation of `BalancedBrackets.solve`. This method receives a string that contains only the characters: `(`, `)`, `{`, `}`, `[`, `]`, and space. It determines if the string contains a valid sequence of parentheses and brackets, that is, every opening bracket has a closing bracket even if they are nested. [**12 pts**]

Examples:

- `( ) [ ]` → true
- `( [ ) ]` → false
- `( { () () } )` → true
- `( ]` → false

Hint: you can solve this problem efficiently using a stack. Go over the characters in the input string.

- If the current character is an opening bracket (`(` or `{` or `[`), then push it to stack.
- If the current character is a closing bracket (`)` or `}` or `]`) then pop from the stack. If the popped character is the matching opening bracket, then fine else, brackets are not balanced.

Notice in `BalancedBrackets.solve` we declared a stack using the Java's built-in `Stack`. The built-in stack has the methods `pop`, `push`, and `size` similar to the Stack ADT declared in an earlier question.