<table>
<tr><th>N</th><th>Problem Statement</th><th>Difficulty</th></tr>
<tr><td>1</td><td>Given a string, reverse it using a stack.

<table><tr><th>Sample Input</th><th>Sample Output</th></tr><tr><td>hello</td><td>olleh</td></tr><tr><td>world</td><td>dlrow</td></tr><tr><td>racecar</td><td>racecar</td></tr></table></td><td>*</td></tr>
<tr><td>2</td><td>Given a string, find the count of all characters in the string. The count would be case sensitive, meaning that if the word contains an uppercase and lowercase character of the same letter, they must be handled separately.

<table><tr><th>Sample Input</th><th>Sample Output</th></tr><tr><td>hello</td><td>h -> 1
e -> 1
l -> 2
o -> 1</td></tr><tr><td>WworlD</td><td>W -> 1
w -> 1
o -> 1
r -> 1
l -> 1
D -> 1</td></tr></table></td><td>*</td></tr>
<tr><td>3</td><td>Given a string, find if the string is a palindrome using nothing but the stack data structure.

<table><tr><th>Sample Input</th><th>Sample Output</th></tr><tr><td>hello</td><td>Not a palindrome</td></tr><tr><td>racecar</td><td>Palindrome</td></tr></table></td><td>**</td></tr>
<tr><td>4</td><td>Given a string containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:
1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.</td><td>**</td></tr>
</table>

3. Every close bracket has a corresponding open bracket of the same type.

Sample Input	**Sample Output**
()	True
()[]{}	True
(]	False
([)]	False