Problem Challenge 2

Comparing Strings containing Backspaces (medium)

Given two strings containing backspaces (identified by the character '#'), check if the two strings are equal.

Example 1:

```
Input: str1="xy#z", str2="xzz#"
Output: true
Explanation: After applying backspaces the strings become "xz" and "xz" respectively.
```

Example 2:

```
Input: str1="xy#z", str2="xyz#"
Output: false
Explanation: After applying backspaces the strings become "xz" and "xy" respectively.
```

Example 3:

```
Input: str1="xp#", str2="xyz##"
Output: true
Explanation: After applying backspaces the strings become "x" and "x" respectively.
In "xyz##", the first '#' removes the character 'z' and the second '#' removes the characte
r 'y'.
```

Example 4:

```
Input: str1="xywrrmp", str2="xywrrmu#p"
Output: true
Explanation: After applying backspaces the strings become "xywrrmp" and "xywrrmp" respectively.
```

Try it yourself

Try solving this question here:

🔸 Java	🐍 Python3	JS JS	⊕ C++

```python
1  def backspace_compare(str1, str2):
2    return create_new_str(str1) == create_new_str(str2)
3
4  def create_new_str(str1):
5    stack = []
6    new_string = ""
7
8    for s in str1:
9      stack.append(s)
10
11   while len(stack) > 0:
12     top = stack[-1]
13     if top == "#":
14       backspaces = 1
15       stack.pop()
16
17       # Find all backspaces
18       while len(stack) > 0:
19         currTop = stack[-1]
20         if currTop == "#":
```

```
21            backspaces += 1
22            stack.pop()
23          else:
24            break
25
26        # Pop all numbers after the backspaces
27        while backspaces > 0:
28          stack.pop()
29          backspaces -= 1
30
31        if len(stack) > 0:
32          top = stack[-1]
33        else:
34          break
35
36    new_string += top
37    stack.pop()
38
39    return new_string
40
41
```

Test   Save   Reset   ⛶

← Back

Next →

✔ Completed

⊘ Report an Issue   ? Ask a Question