

LC 394. Decode String

Question

Given an encoded string, return its decoded string.

The encoding rule is: `k[encoded_string]`, where the *encoded_string* inside the square brackets is being repeated exactly *k* times.

Note that *k* is guaranteed to be a positive integer.

You may assume that the input string is always valid; No extra white spaces, square brackets are well-formed, etc.

Furthermore, you may assume that the original data does not contain any digits and that digits are only for those repeat numbers, *k*.

For example, there won't be input like `3a` or `2[4]`.

Examples:

```
s = "3[a]2[bc]", return "aaabcbc".
s = "3[a2[c]]", return "accaccacc".
s = "2[abc]3[cd]ef", return "abcabccdcdcdef".
```

Solution

```
class Solution:
    def decodeString(self, s: str) -> str:
        #Solution
        if not s:
            return ''
        stack = ['']
        for ch in s:
            if ch.isdigit():
                if stack[-1].isdigit():
                    stack[-1] += ch
                else:
                    stack.append(ch)
            elif ch == '[':
                stack.append('')
            elif ch == ']':
                cur = stack.pop()
                count = stack.pop()
                stack[-1] += cur * int(count)
            else:
                stack[-1] += ch
        return stack[0]
```