5617. Goal Parser Interpretation

My Submissions (/contest/weekly-contest-218/problems/goal-parser-interpretation/submissions/)   Back to Contest (/contest/weekly-contest-218/)

You own a **Goal Parser** that can interpret a string `command` . The `command` consists of an alphabet of `"G"` , `"()"` and/or `"(al)"` in some order. The Goal Parser will interpret `"G"` as the string `"G"` , `"()"` as the string `"o"` , and `"(al)"` as the string `"al"` . The interpreted strings are then concatenated in the original order.

Given the string `command` , return *the* **Goal Parser**'s *interpretation of* `command` .

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Easy

Example 1:

```
Input: command = "G()(al)"
Output: "Goal"
Explanation: The Goal Parser interprets the command as follows:
G -> G
() -> o
(al) -> al
The final concatenated result is "Goal".
```

Example 2:

```
Input: command = "G()()()()(al)"
Output: "Gooooal"
```

Example 3:

```
Input: command = "(al)G(al)()()G"
Output: "alGalooG"
```

Constraints:

- `1 <= command.length <= 100`
- `command` consists of `"G"` , `"()"` , and/or `"(al)"` in some order.

JavaScript ▾

```javascript
/**
 * @param {string} command
 * @return {string}
 */
const interpret = (command) => {
    let n = command.length;
    let res = "";
    for (let i = 0; i < n;) {
        if (command[i] == 'G') {
            res += 'G';
            i++;
        } else if (command[i] == '(') {
            if (i + 1 < n) {
                if (command[i + 1] == ')') {
                    res += 'o';
                    i += 2;
                } else {
                    res += 'al';
                    i += 4;
                }
            }
        }
    }
    return res;
};
```

☐ **Custom Testcase**   Use Example Testcases

Submission Result: Accepted (/submissions/detail/427639852/) ❓

More Details ❯ (/submissions/detail/427639852/)

Share your acceptance!

◀ 1