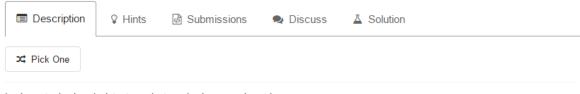
227. Basic Calculator II



Implement a basic calculator to evaluate a simple expression string.

The expression string contains only **non-negative** integers, +, -, *, / operators and empty spaces . The integer division should truncate toward zero.

Example 1:

```
Input: "3+2*2"
Output: 7
```

Example 2:

```
Input: " 3/2 "
Output: 1
```

Example 3:

```
Input: " 3+5 / 2 "
Output: 5
```

Note:

- · You may assume that the given expression is always valid.
- Do not use the eval built-in library function.

```
5 public class L227 {
        public int calculate(String s) {
7
8
            if(s == null || s.length() == 0) {
9
                return 0;
.0
            Stack<Integer> stk = new Stack<>();
.1
.2
            int num = 0, res = 0;
.3
            char lastSign = '+';
.4
            char [] cArray = s.toCharArray();
.5
.6
            for(int i = 0; i < cArray.length; i ++) {</pre>
.7
                char c = cArray[i];
                if(c >= '0' && c <= '9') {
.8
.9
                    //这个是怕num不止一位数
10
                    num = num * 10 + c - '0';
11
12
                if(c == '+' || c == '-' || c == '*' || c == '/' || i == cArray.length - 1) {
                    if(lastSign == '+' || lastSign == '-') {
!3
4
                        int temp = lastSign == '+' ? num : -num;
15
                        stk.push(temp);
16
                        res += temp;
17
                    //如果上一个是乘法或者是除法,那么把上次结果弹出,计算后再加入
18
                    if(lastSign == '/' || lastSign == '*') {
19
                        res -= stk.peek();
0
                        int temp = lastSign == '*' ? stk.pop() * num : stk.pop() / num;
1
12
                        stk.push(temp);
13
                        res += temp;
4
                    lastSign = c;
15
                    num = 0;
16
17
                }
19
           return res;
-0
        }
1 }
```