

棒球比赛

你现在是一场采用特殊赛制棒球比赛的记录员。这场比赛由若干回合组成，过去几回合的得分可能会影响以后几回合的得分。

比赛开始时，记录是空白的。你会得到一个记录操作的字符串列表 `ops`，其中 `ops[i]` 是你需要记录的第 `i` 项操作，`ops` 遵循下述规则：

1. 整数 `x` - 表示本回合新获得分数 `x`
2. `"+"` - 表示本回合新获得的得分是前两次得分的总和。题目数据保证记录此操作时前面总是存在两个有效的分数。
3. `"D"` - 表示本回合新获得的得分是前一次得分的两倍。题目数据保证记录此操作时前面总是存在一个有效的分数。
4. `"C"` - 表示前一次得分无效，将其从记录中移除。题目数据保证记录此操作时前面总是存在一个有效的分数。

请你返回记录中所有得分的总和。

示例 1：

输入：`ops = ["5","2","C","D","+"]`

输出：`30`

解释：

`"5"` - 记录加 5 ，记录现在是 [5]

`"2"` - 记录加 2 ，记录现在是 [5, 2]

`"C"` - 使前一次得分的记录无效并将其移除，记录现在是 [5].

`"D"` - 记录加 $2 * 5 = 10$ ，记录现在是 [5, 10].

`"+"` - 记录加 $5 + 10 = 15$ ，记录现在是 [5, 10, 15].

所有得分的总和 $5 + 10 + 15 = 30$

示例 2：

输入：`ops = ["5","-2","4","C","D","9","+","+"]`

输出：`27`

解释：

`"5"` - 记录加 5 ，记录现在是 [5]

"-2" - 记录加 -2 ，记录现在是 [5, -2]

"4" - 记录加 4 ，记录现在是 [5, -2, 4]

"C" - 使前一次得分的记录无效并将其移除，记录现在是 [5, -2]

"D" - 记录加 $2 * -2 = -4$ ，记录现在是 [5, -2, -4]

"9" - 记录加 9 ，记录现在是 [5, -2, -4, 9]

"+" - 记录加 $-4 + 9 = 5$ ，记录现在是 [5, -2, -4, 9, 5]

"+" - 记录加 $9 + 5 = 14$ ，记录现在是 [5, -2, -4, 9, 5, 14]

所有得分的总和 $5 + -2 + -4 + 9 + 5 + 14 = 27$

示例 3:

输入: ops = ["1"]

输出: 1

```
class Solution {
public:
    int calPoints(vector<string>& ops) {
        vector<int> res;
        int sum=0;
        for(int i=0;i<ops.size();i++)
        {
            if(ops[i]=="C"&&res.size()!=0)
            {
                res.pop_back();
            }
            else if(ops[i]=="D"&&res.size()!=0)
            {
                res.push_back(2*res[res.size()-1]);
            }
            else if(ops[i]=="+"&&res.size()>=2)
            {
                res.push_back(res[res.size()-2]+res[res.size()-1]);
            }
            else
                res.push_back(stoi(ops[i]));
        }
        for(int i=0;i<res.size();i++)
        {
            sum+=res[i];
        }
    }
};
```

```
    }  
    return sum;  
}  
};
```