

題目、610

✓ 610. 邪惡水族箱

Submit Status Discussion / Solution Ranklist Back to Problems List

TopCoder timeflew

User's AC Ratio 100.0% (28/28) Submission's AC Ratio 22.5% (38/169)

Tags apsc apsc-recursion Show solution-related tags

Description

水族箱像大海一樣，是非常冷酷無情的。魚吃飼料，魚吃魚是水中世界裡面絕對的規則。現在呢，就是想要模擬水族箱內的情況。為了簡化問題，水族箱內只有飼料和魚兩種東西，而各自都有各自的「飽滿值」。並且如果一隻魚 a 依序吃了飽滿度為 x, y, z 的東西之後，其飽滿度變成 k 的話，則記為 $a(x, y, z) = k$ 。飽滿值的規則如下：

- 對於飼料來說，它的飽滿度就是一個數字 x 。
- 魚有三種，分別分為 f 魚、 g 魚、與 h 魚（分別簡稱 f, g, h ）。 f 只能吃一個東西，且它的規則為 $f(x) = (2x - 3) \bmod 998244353$ 。 g 魚可以吃兩樣東西，它的規則為 $g(x, y) = (2x + y - 7) \bmod 998244353$ 。 h 魚吃得最多，可以吃三樣東西，且它吃的規則為 $h(x, y, z) = (3x - 2y + z) \bmod 998244353$ 。請注意，吃東西的順序是有差的。

我們可以將一天下來的相食關係表示為一串函數合成，如 $h(4, g(2, 4), f(g(3, 4)))$ 或 $f(f(f(4)))$ 。現在，我們將所有的逗號和括弧都替換成空白輸入，請算出最後一條魚的飽滿值吧！

TLDR：給你三個函數 $f(x) = 2x - 3$ 、 $g(x, y) = 2x + y - 7$ 、 $h(x, y, z) = 3x - 2y + z$ ，和一個去除括弧的函數合成式，請求這個式子的計算結果 $\bmod 998244353$ 。請特別注意取模後的值需要介於 $[0, 998244353]$ 。

Input Format
輸入有兩行。第一行為一個數字 N 代表項數。接下來，會有 N 個以空白分隔的符號，可能為 f 、 g 、 h ，或一個絕對值不超過 1000 的整數。

- $1 \leq N \leq 1500$
- 輸入是一個合法的式子（e.g. 不會有 $g 2$ 這種輸入）。

Output Format
請輸出一個數字，代表答案。

Sample Input 1 7
g f 6 h 7 8 1

Sample Input 2 8
g g 7 1 g 2 f 2

Hints

Sample Output 1 17

Sample Output 2 7

Problem Source
改自 APSC201902

這一題需要考慮函式、遞迴、及字串

Code

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4 #define MOD 998244353
5 int f(int a){
6     return (2 * a - 3 + MOD) % MOD;
7 }
8 int g(int a, int b){
9     return (2 * a + b - 7 + MOD) % MOD;
10 }
11 int h(int a, int b, int c){
12     return (3 * a - 2 * b + c + MOD) % MOD;
13 }
14 int main() {
15     int n;
16     cin >> n;
17     char *arr = new char [n + 1];
18     for (int i = 0; i < n; i++){
19         cin >> arr[i];
20     }
21     while (n > 1){
22         for (int i = n - 1; i >= 0; i--){
23             if (arr[i] == 'f'){
24                 int a = arr[i + 1] - '0';
25                 arr[i] = f(a) + '0';
26                 for (int j = i + 1; j < n; j++){
27                     arr[j] = arr[j + 1];
28                 }
29                 n--;
30             }
31             if (arr[i] == 'g'){
32                 int a = arr[i + 1] - '0';
33                 int b = arr[i + 2] - '0';
34                 arr[i + 1] = g(a, b) + '0';
35                 for (int j = i + 1; j < n; j++){
36                     arr[j] = arr[j + 1];
37                 }
38                 n -= 2;
39             }
40             if (arr[i] == 'h'){
41                 int a = arr[i + 1] - '0';
42                 int b = arr[i + 2] - '0';
43                 int c = arr[i + 3] - '0';
44                 arr[i + 1] = h(a, b, c) + '0';
45                 for (int j = i + 1; j < n; j++){
46                     arr[j] = arr[j + 1];
47                 }
48                 n -= 3;
49             }
50         }
51     }
52     cout << arr[0];
53     return 0;
54 }
```

一開始我的想法是，先找到‘f’的位置，然後再一次次的把陣列往前，但這樣做的問題是，函式‘f’，‘g’，‘h’後面不一定是數字，所以還要先判斷這個字元的數字是否為數字。

第二個問題是，這樣做會無法處理多位數的問題，如果可以利用遞迴將能減少題目的複雜度

Problem	Total Time (ms)	Max Memory (RSS, KiB) ⓘ	Verdict ⓘ	Score		
610. 那恩水旗箱	71	3456	Wrong Answer	0		
Subtask Results	Testdata Range	Constraints		Score		
1	0~1	範例測資		0 / 0		
2	0~16	無額外限制		0 / 100		
Testdata Results	Subtasks	Time (ms)	Memory (VSS, KiB) ⓘ	Memory (RSS, KiB) ⓘ	Verdict ⓘ	Score
0	① ②	3.1	6364	3456	Accepted	100
1	① ②	4.1	6364	3456	Accepted	100
2	②	4.2	6364	3456	Wrong Answer	0
3	②	4.1	6364	3456	Accepted	100
4	②	4.0	6364	3456	Wrong Answer	0
5	②	4.1	6364	3456	Wrong Answer	0
6	②	4.6	6364	3456	Wrong Answer	0
7	②	4.4	6364	3456	Accepted	100
8	②	4.5	6364	3456	Wrong Answer	0
9	②	4.5	6364	3456	Wrong Answer	0
10	②	4.0	6364	3456	Accepted	100
11	②	4.0	6364	3456	Accepted	100
12	②	3.9	6364	3456	Accepted	100
13	②	4.5	6364	3456	Wrong Answer	0
14	②	4.4	6364	3456	Wrong Answer	0
15	②	4.6	6364	3456	Wrong Answer	0
16	②	4.3	6364	3456	Wrong Answer	0

```

1 #include<iostream>
2 using namespace std;
3 #define MOD 998244353
4 char c[1600];
5 int num[1600];
6
7 int i;
8 int opr() {
9     if (c[i] == 'f') {
10         i++;
11         int x = opr();
12         return (x * 2 - 3 + MOD) % MOD;
13     } else if (c[i] == 'g') {
14         i++;
15         int x = opr();
16         int y = opr();
17         return (x * 2 + y - 7 + MOD) % MOD;
18     } else if (c[i] == 'h') {
19         i++;
20         int x = opr();
21         int y = opr();
22         int z = opr();
23         return (x * 3 - y * 2 + z + MOD*2) % MOD;
24     } else {
25         i++;
26         return num[i-1];
27     }
28 }
29 int main(){
30 int n;
31     cin >> n;
32     for (int i = 0 ; i < n ; i++) {
33         char s[6];
34         cin >> s;
35         if (s[0] >= '0' && s[0] <= '9' || s[0] == '-') {
36             num[i] = 0;
37             bool ng = 0;
38             if (s[0] == '-') ng = 1;
39             for (int j = 0 ; s[j] != '\0' ; j++) {
40                 if (s[j] == '-') continue;
41                 num[i] = num[i] * 10 + s[j] - '0';
42                 //cout << s[j] << " " << (int)num[i] << "\n";
43             }
44             if (ng) num[i] *= -1;
45             c[i] = 0;
46         } else {
47             c[i] = s[0];
48         }
49     }
50     i = 0;
51     int ans = opr();
52     cout << ans << '\n';
53 }

```

接著我利用遞迴得方式

當輸入 $g\ f\ 3\ 4\ 5$ 時

`opr()`

$\rightarrow c[0] = 'g'$

$\rightarrow \text{opr}() \text{ for } x$

$\rightarrow c[1] = 'f'$

$\rightarrow \text{opr}() \text{ for } x$

$\rightarrow c[2] = 0 \rightarrow \text{回傳 } \text{num}[2] = 3$

$\rightarrow \text{回傳 } (3*2 - 3) = 3$

$\rightarrow \text{opr}() \text{ for } y$

$\rightarrow c[3] = 0 \rightarrow \text{回傳 } \text{num}[3] = 4$

$\rightarrow g(3,4) = 2*3 + 4 - 7 = 3$

但這樣依然有部分測資沒有過

Results of submission #70163						
Back to Submissions List						
Problem	Total Time (ms)		Max Memory (RSS, KiB) ⓘ	Verdict ⓘ	Score	
610. 邻恩水族箱	71		3456	Accepted	100	
Subtask Results						
Subtask no.	Testdata Range	Constraints			Score	
1	0~1	範例測資			0 / 0	
2	0~16	無額外限制			100 / 100	
Testdata Results						
Testdata no.	Subtasks	Time (ms)	Memory (VSS, KiB) ⓘ	Memory (RSS, KiB) ⓘ	Verdict ⓘ	Score
0	① ②	2.7	6368	3456	Accepted	100
1	① ②	4.0	6368	3456	Accepted	100
2	②	4.1	6368	3456	Accepted	100
3	②	4.0	6368	3456	Accepted	100
4	②	4.2	6368	3456	Accepted	100
5	②	4.1	6368	3456	Accepted	100
6	②	4.7	6368	3456	Accepted	100
7	②	4.5	6368	3456	Accepted	100
8	②	4.5	6368	3456	Accepted	100
9	②	4.4	6368	3456	Accepted	100
10	②	4.1	6368	3456	Accepted	100
11	②	3.9	6368	3456	Accepted	100
12	②	4.0	6368	3456	Accepted	100
13	②	4.5	6368	3456	Accepted	100
14	②	4.5	6368	3456	Accepted	100
15	②	4.5	6368	3456	Accepted	100
16	②	4.4	6368	3456	Accepted	100

```

1 #include<iostream>
2 using namespace std;
3 #define MOD 998244353
4
5 char c[1600];
6 long long num[1600]; // 改 long long 防溢出
7 int i;
8
9 long long opr() {
10     if (c[i] == 'f') {
11         i++;
12         long long x = opr();
13         return (2 * x - 3 + MOD) % MOD;
14     } else if (c[i] == 'g') {
15         i++;
16         long long x = opr();
17         long long y = opr();
18         return (2 * x + y - 7 + MOD * 2) % MOD;
19     } else if (c[i] == 'h') {
20         i++;
21         long long x = opr();
22         long long y = opr();
23         long long z = opr();
24         return ((3 * x - 2 * y + z) % MOD + MOD) % MOD; // 確保非負
25     } else {
26         i++;
27         return (num[i - 1] % MOD + MOD) % MOD;
28     }
29 }
30
31 int main() {
32     int n;
33     cin >> n;
34     for (int i = 0 ; i < n ; i++) {
35         char s[20];
36         cin >> s;
37         if (((s[0] >= '0' && s[0] <= '9') || s[0] == '-') && (s[0] <= '9' || s[0] == '0')) {
38             num[i] = 0;
39             bool ng = 0;
40             if (s[0] == '-') ng = 1;
41             for (int j = 0 ; s[j] != '\0' ; j++) {
42                 if (s[j] == '-') continue;
43                 num[i] = num[i] * 10 + s[j] - '0';
44             }
45             if (ng) num[i] *= -1;
46             c[i] = 0;
47         } else {
48             c[i] = s[0];
49         }
50     }
51     i = 0;
52     long long ans = opr();
53     cout << ans % MOD << '\n';
54 }

```

第三次嘗試，我將字串中純數字的 `num[]` 也做了 `mod` 處理，因為處理後的數字必為正數

以下為程式的解析

第一步是要先判斷 `tokens` 類型

```
if (((s[0] >= '0' && s[0] <= '9') || s[0] == '-')) //若第一個字元是數字或負號，就當作數字。
```

若是數字，先判斷是否為負數，再把字串轉成整數存入 `num[i]`，接著設 `c[i] = 0` 表示這個位置是數字。

如果是字母就 `c[i] = s[0];`

```
long long ans = opr(); //這裡 opr() 會從 c[0] 開始，且因為 num[]、c[]、int i，都是全域變數，所以函數就不用加變數
```