

E5-H题

题目描述

妮妮发现了一个剪刀石头布的新玩法，初始给定一个长度为 n 的剪刀石头布序列 S ，其中只有 S,R,P 三种字符，分别表示剪刀、石头与布。游戏会进行 $n-1$ 轮，每一轮会根据上一轮的序列生成一个新的剪刀石头布序列，每一轮操作按如下方式进行：

- 设上一轮的序列（若为第一轮则为初始给定的序列）为 S ，这一轮生成的新序列为 T ，有 $|T|=|S|-1$ 。
- 对于 $i=1,2,\dots,n-1$ ，如果 $S_i = S_{i+1}$ 则有 $T_i=S_i$ ，否则 T_i 为 S_i 和 S_{i+1} 进行普通剪刀石头布游戏获胜的一方所对应的字符。

妮妮发现最后一轮所生成的序列仅包含一个字符，她想知道具体是哪个字符。

输入格式

第一行一个正整数 t ($1 \leq t \leq 50$)，表示数据组数。

对于每组数据，一行一个由 S,R,P组成的字符串 S ($1 \leq |S| \leq 10^6$)，含义同题目描述。

对于所有数据，保证 $|S|$ 之和不超过 3×10^6 。

输出格式

对于每组数据，输出一行一个字符，表示最后一轮生成序列所包含的字符。

输入样例

```
1 1
2 SRP
```

输出样例

```
1 P
```

样例解释

SRP→RP→P

解题思路

先观察以下两个例子：

1. RSPRSP --> R
2. PSRPSR --> R

由例子1和2可以看出，当给出的字符串 s 是“单调的”，最终结果是“最大”的那个字符。

于是，我们可以构建一个单调栈，维护栈中元素满足：第 i 个元素 **大于** 第 $i+1$ 个元素，即第 i 个字符能**赢过** 第 $i+1$ 个字符。（ $i=0, 1, 2, \dots$ ）

要构建的单调栈举例如下：

RSPRSP

其中， $R > S > P > R > S > P$

接下来，我们只需要让 s 中的字符依次入栈：

1. 当 $S[i]$ **小于** 栈顶字符时，直接入栈；
2. 当 $S[i]$ **大于或等于** 栈顶字符时，栈顶字符出栈，然后不断和当前栈顶元素进行比较；直到 $S[i]$ **小于** 当前栈顶字符，让 $S[i]$ 入栈

最后栈底元素就是所求答案。

参考代码

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  typedef long long LL;
6
7  const int SIZEN = 1e6 + 50;
8  using namespace std;
9
10 char s[SIZEN];
11 char st[SIZEN];
12 int top = 0;
13
14 int fun(char a, char b) {
15     if (a == b) return 0;
16     if (a == 'S' && b == 'P') return 1;
17     if (a == 'R' && b == 'S') return 1;
18     if (a == 'P' && b == 'R') return 1;
19     return -1;
20 }
21
22
23 int main() {
24     int tt;
```

```

25     cin >> tt;
26
27     while (tt--) {
28         memset(s, 0, sizeof(s));
29         memset(st, 0, sizeof(st));
30         scanf("%s", s);
31         top = 0;
32         for (int i = 0; i < strlen(s); i++) {
33             if (top == 0 || fun(s[i], st[top - 1]) == -1) {
34                 st[top++] = s[i];
35             }
36             else if (fun(s[i], st[top - 1]) == 0) {
37                 continue;
38             }
39             else if (fun(s[i], st[top - 1]) == 1) {
40                 st[top - 1] = s[i];
41                 while ((fun(s[i], st[top - 2]) == 1 || fun(s[i], st[top - 2]) ==
0) && top > 1) {
42                     st[top - 2] = s[i];
43                     st[top - 1] = '\0';
44                     top--;
45                 }
46             }
47         }
48         printf("%c\n", st[0]);
49     }
50 }

```