

题目描述

幼儿园两个班的小朋友在排队时混在了一起。每位小朋友都知道自己是否与前一位小朋友同班。请你帮忙把混班的小朋友找出来。

小朋友的编号是整数，与前一位小朋友间用#表示，不同班用N表示。

学生序号范围(0 999)。如果输入不合法则打印ERROR。

输入描述

输入为空格分开的小朋友编号和是否同班标志。

输出描述

输出为两行，每一行记录一个班小朋友的编号，编号用空格分开，且：

- 1. 编号按顺序排列。
- 2. 若只有一个班的小朋友，第二行为空行。

用例

输入	1N 2Y 3N 4Y
输出	1 2 3 4
说明	2B的同班标志为Y，因此和1同班。 3B的同班标志为N，因此和1 2不同班。 4B的同班标志为Y，因此和3同班。 所以1、2同班、3、4同班。输出为 1 2 3 4
输入	1N 2Y 3N 4Y 5Y
输出	1 2 3 4 5
说明	无

题目解析

这题解起来可能和这类型的：

首先将输入的字符串按“#”拆分为数组arr，然后将每个数组元素按“/”分隔得到每个学生的序号和班级信息。

得到学生序号后，判断是否在(0 999)范围内，若不在则返回ERROR，程序终止。

如果学生序号符合要求，则定义一个res数组用来存储学生的班级信息。假设arr[i]学生的班级是true班，然后从1开始遍历arr学生班级信息，判断arr[i]的值是否N，若是，则res[i]设置为res[i-1]，若不是，则res[i]设置为res[i-1]的同班标志。

这样res中就存储了两种信息，true和false，我们将为true的元素的索引取出来+1，就是一个班的序号，将false的元素索引取出来+1，就是另一个班的学生序号。

Java算法源码

```
1 import java.util.Scanner;
2 import java.util.StringTokenizer;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         String[] arr = sc.nextLine().split("#");
8         getResult(arr);
9     }
10
11     public static void getResult(String[] arr) {
12         for (int i = 0; i < arr.length; i++) {
13             StringTokenizer tmp = arr[i].split("/");
14
15             int idx = Integer.parseInt(tmp[0]);
16             String isSame = tmp[1];
17
18             if (idx <= 0 || idx >= 999) {
19                 System.out.println("ERROR");
20                 return;
21             }
22
23             arr[i] = isSame;
24         }
25
26         boolean[] res = new boolean[arr.length];
27         res[0] = true;
28
29         for (int i = 1; i < arr.length; i++) {
30             if ("N".equals(arr[i])) {
31                 res[i] = !res[i - 1];
32             } else {
33                 res[i] = res[i - 1];
34             }
35         }
36
37         StringTokenizer one = new StringTokenizer(" ");
38         StringTokenizer two = new StringTokenizer(" ");
39
40         for (int i = 0; i < res.length; i++) {
41             if (res[i] == true) {
42                 one.add(i + 1 + " ");
43             } else {
44                 two.add(i + 1 + " ");
45             }
46
47             System.out.println(one);
48             System.out.println(two);
49         }
50     }
51 }
```

JS算法源码

```
1 // JavaScript 版本 实现与 Java 类似
2 const readline = require("readline");
3
4 const rl = readline.createInterface({
5     input: process.stdin,
6     output: process.stdout,
7 });
8
9 rl.on("line", (line) => {
10     getResult(line.split("#"));
11 });
12
13 function getResult(arr) {
14     for (let i = 0; i < arr.length; i++) {
15         let [idx, isSame] = arr[i].split("/");
16
17         if ((idx <= 0 || idx >= 999)) {
18             console.log("ERROR");
19             return;
20         }
21
22         arr[i] = isSame;
23     }
24
25     const res = new Array(arr.length).fill(true);
26
27     for (let i = 1; i < arr.length; i++) {
28         if ("N" === arr[i]) {
29             res[i] = !res[i - 1];
30         } else {
31             res[i] = res[i - 1];
32         }
33     }
34
35     const one = [];
36     const two = [];
37
38     for (let i = 0; i < res.length; i++) {
39         if (res[i]) one.push(i + 1);
40         else two.push(i + 1);
41     }
42
43     console.log(one.join(" "));
44     console.log(two.join(" "));
45 }
```

Python算法源码

```
1 # 与JS类似
2 arr = input().split()
3
4 # 定义函数
5 def getResult():
6     for i in range(len(arr)):
7         tmp = arr[i].split("/")
8
9         idx = int(tmp[0])
10         isSame = tmp[1]
11
12         if idx <= 0 or idx >= 999:
13             print("ERROR")
14             return
15
16         arr[i] = isSame
17
18     res = [True]*len(arr)
19
20     for i in range(1, len(arr)):
21         if arr[i] == "N":
22             res[i] = not res[i-1]
23         else:
24             res[i] = res[i-1]
25
26     one = []
27     two = []
28
29     for i in range(len(res)):
30         if res[i]:
31             one.append(i+1)
32         else:
33             two.append(i+1)
34
35     print(" ".join(map(str, one)))
36     print(" ".join(map(str, two)))
37
38 # 调用函数
39 print(getResult())
```