

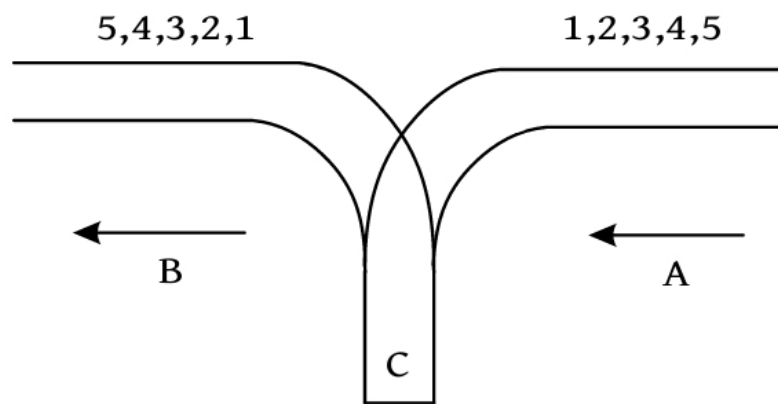
## Rails

(rails.cpp/.c)

限制：1S 256MB

### 题目描述：

某城市有一个火车站，铁轨铺设如下图所示。有  $n$  ( $n \leq 1000$ ) 节车厢从 A 方向驶入车站，将其按进站的顺序编号为  $1 \sim n$ 。你的任务是判断是否能让它们按照某种特定的顺序进入 B 方向的铁轨并驶出车站。例如，出栈顺序 (5 4 1 2 3) 是不可能的，但出栈顺序 (5 4 3 2 1) 是可能的。为了重组车厢，你可以借助中转站 C。中转站 C 是一个可以停放任意多节车厢的车站，但由于末端封顶，驶入 C 的车厢必须按照相反的顺序驶出 C。对于每节车厢，一旦从 A 移入 C，就不能返回 A 了；一旦从 C 移入 B，就不能返回 C 了。在任意时刻只有两种选择：A 到 C 和 C 到 B。



### 输入：(rails.in)

输入包含多组数据，对于每一组数据，第 1 行是一个整数  $n$ 。接下来的若干行，每行  $n$  个数，代表  $1 \sim n$  车厢的出栈顺序，最后一行只有一个整数 0。

最后一组数据 “ $n=0$ ”，输入结束，不输出答案。

### 输出：(rails.out)

对每行的出栈顺序都单行输出 “Yes” 或 “No”。对每组数据都在最后输出空行。

#### 输入样例

```
5
1 2 3 4 5
5 4 1 2 3
0
6
6 5 4 3 2 1
0
0
```

#### 输出样例

```
Yes
No
Yes
```

## 题解

本题中的 C 就是一个栈，1~n 车厢按顺序依次从 A 端进来，首先和 B 端的字符进行比较，如果相等，则直接从 B 端出去，如果不相等则进入栈 C。如果栈非空，则判断栈顶元素是否与 B 端的字符相等，如果相等则出栈，一直比较下去。如果 1~n 车厢都已处理完毕，B 端字符还未处理完，则输出 “No”，否则输出 “Yes”。

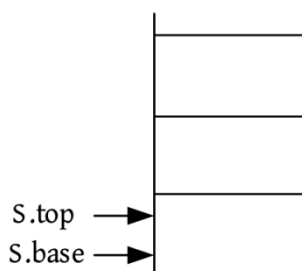
需要特别注意：输入包含多组数据，每组数据都以 0 结束，每组数据输出结束时都会加一个空行。最后一组数据为 0，不输出。

### 1. 算法设计

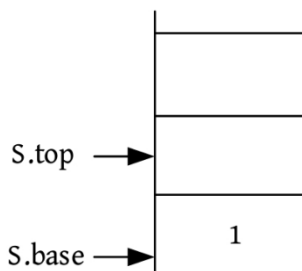
- (1) 输入 n，如果 n 为 0，则结束。
- (2) 输入第 1 组数据的第 1 个字符。
- (3) 如果 B[1]不为 0，则读入余下的字符并将其存入 B[]。
- (4) 初始化一个栈 s。
- (5) 1~n 车厢依次与 B 端的字符进行比较，如果相等，则直接从 B 端出栈，否则入栈。
- (6) 如果栈非空，则判断栈顶元素是否与 B 端的字符相等，相等则出栈，一直比较下去。
- (7) 如果 1~n 车厢都已处理完毕，B 端字符还未处理完，则输出 “No”，否则输出 “Yes”。

### 2. 图解

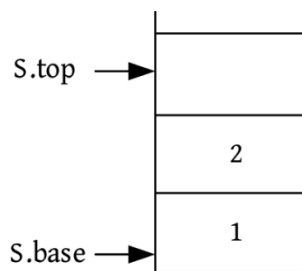
- (1) 以输入 3 2 1 5 4 为例，将序列存入 B[]，j=1，初始化一个栈。



- (2) i=1，将 i 与 B[1]=3 进行比较，不相等，1 入栈。

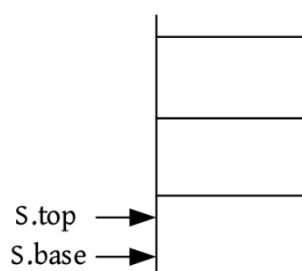


(3)  $i=2$ , 将  $i$  与  $B[1]=3$  进行比较, 不相等, 2 入栈。

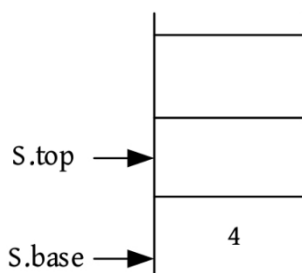


(4)  $i=3$ , 将  $i$  与  $B[1]=3$  进行比较, 相等,  $j++$  ( $j=2$ )。

(5) 栈非空, 栈顶元素 2 和  $B[2]=2$  相等, 出栈,  $j++$  ( $j=3$ ); 栈非空, 栈顶元素 1 和  $B[3]=1$  相等, 出栈,  $j++$  ( $j=4$ ); 此时栈空。

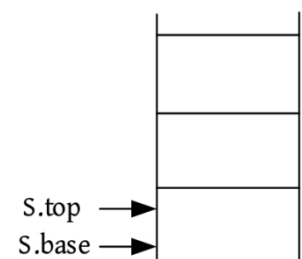


(6)  $i=4$ , 将  $i$  与  $B[4]=5$  进行比较, 不相等, 4 入栈。



(7)  $i=5$ , 将  $i$  与  $B[4]=5$  进行比较, 相等,  $j++$  ( $j=5$ )。

(8) 栈非空, 栈顶元素 4 和  $B[5]=4$  相等, 出栈,  $j++$  ( $j=6$ ); 此时栈空。



(9) 此时  $j > n$ , 输出 “Yes”。

### 3. 算法实现

[rails.cpp](#)