

## 双栈排序

### 【问题描述】

Tom 最近在研究一个有趣的排序问题。如图所示，通过 2 个栈 S1 和 S2，Tom 希望借助以下 4 种操作实现将输入序列升序排序。

操作 a

如果输入序列不为空，将第一个元素压入栈 S1

操作 b

如果栈 S1 不为空，将 S1 栈顶元素弹出至输出序列

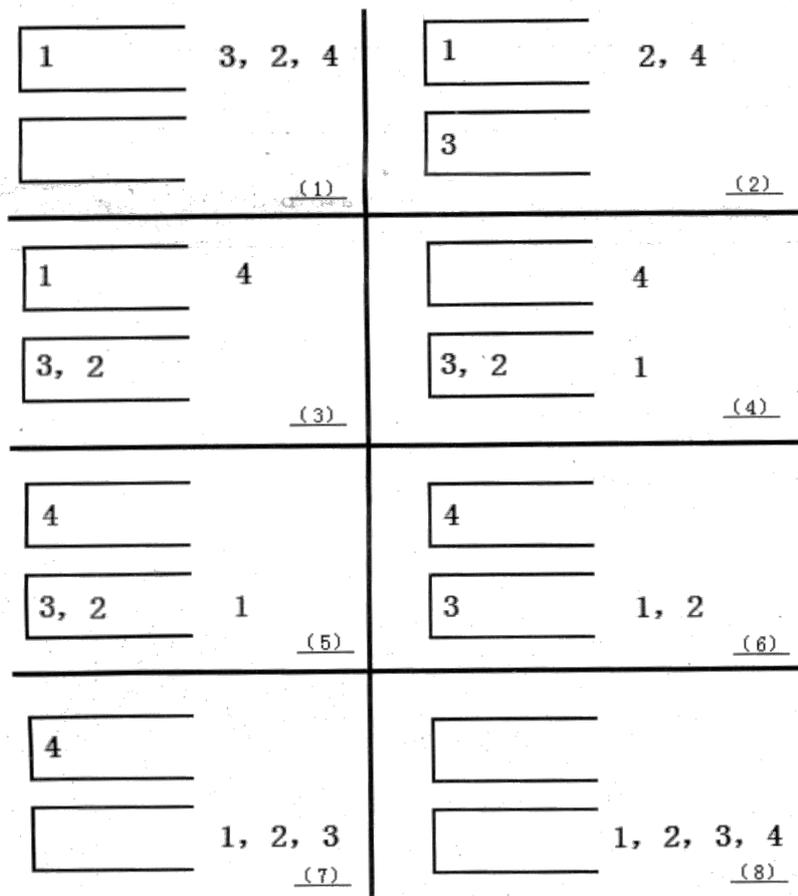
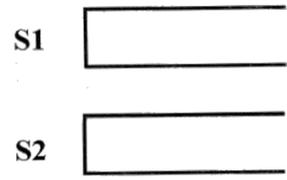
操作 c

如果输入序列不为空，将第一个元素压入栈 S2

操作 d

如果栈 S2 不为空，将 S2 栈顶元素弹出至输出序列

如果一个  $1 \sim n$  的排列 P 可以通过一系列操作使得输出序列为  $1, 2, \dots, (n-1), n$ ，Tom 就称 P 是一个“可双栈排序排列”。例如  $(1, 3, 2, 4)$  就是一个“可双栈排序序列”，而  $(2, 3, 4, 1)$  不是。下图描述了一个将  $(1, 3, 2, 4)$  排序的操作序列： $\langle a, c, c, b, a, d, d, b \rangle$



当然，这样的操作序列有可能有几个，对于上例  $(1, 3, 2, 4)$ ， $\langle a, c, c, b, a, d, d, b \rangle$  是另外一个可行的操作序列。Tom 希望知道其中字典序最小的操作序列是什么。

### 【输入】

输入文件的第一行是一个整数  $n$ 。

第二行有  $n$  个用空格隔开的正整数，构成一个  $1 \sim n$  的排列。

**【输出】**

输出文件共一行，如果输入的排列不是“可双栈排序排列”，输出数字 0；否则输出字典序最小的操作序列，每两个操作之间用空格隔开，行尾没有空格。

**【输入输出样例 1】**

twostack.in	twostack.out
4 1 3 2 4	a b a a b b a b

**【输入输出样例 2】**

twostack.in	twostack.out
4 2 3 4 1	0

**【输入输出样例 3】**

twostack.in	twostack.out
3 2 3 1	a c a b b d

**【限制】**

30%的数据满足：  $n \leq 10$

50%的数据满足：  $n \leq 50$

100%的数据满足：  $n \leq 1000$