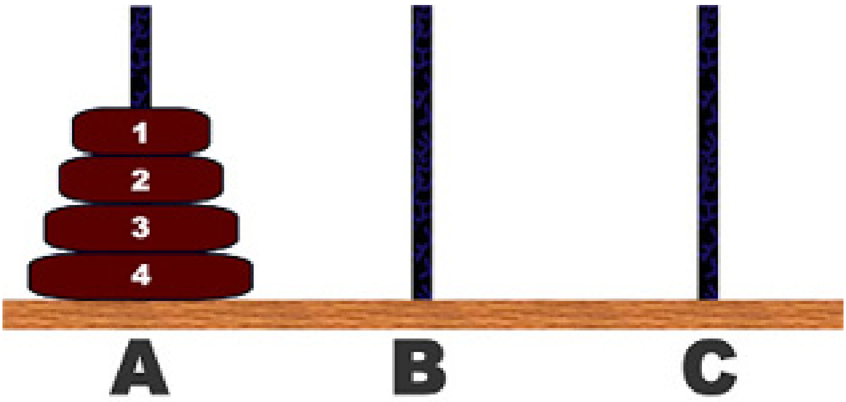
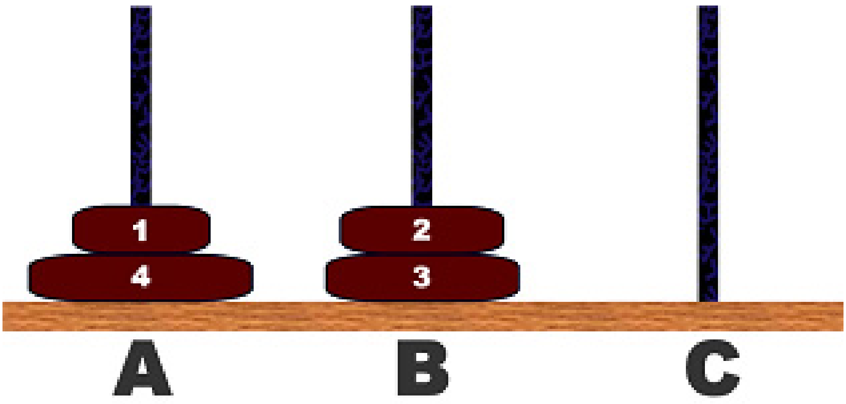
The (Three peg) Tower of Hanoi problem is a popular one in computer science. Briefly the problem is to transfer all the disks from peg-**A** to peg-**C** using peg-**B** as intermediate one in such a way that at no stage a larger disk is above a smaller disk.



Normally, we want the minimum number of moves required for this task. The problem is used as an ideal example for learning recursion. It is so well studied that one can find the sequence of moves for smaller number of disks such as **3** or **4**. A trivial computer program can find the case of large number of disks also.

Here we have made your task little bit difficult by making the problem more flexible. Here the disks can be in any peg initially.



If more than one disk is in a certain peg, then they will be in a valid arrangement (larger disk will not be on smaller ones). We will give you two such arrangements of disks. You will have to find out the minimum number of moves, which will transform the first arrangement into the second one. Of course you always have to maintain the constraint that smaller disks must be upon the larger ones.

Input

The input file contains at most **100** test cases. Each test case starts with a positive integer N ($1 \leq N \leq 60$), which means the number of disks. You will be given the arrangements in next two lines. Each arrangement will be represented by N integers, which are **1**, **2** or **3**. If the i-th ($1 \leq i \leq N$) integer is **1**, you should consider that i-th disk is on Peg-**A**. Input is terminated by $N = 0$. This case should not be processed.

Output

Output of each test case should consist of a line starting with '`Case #: `' where # is the test case number. It should be followed by the minimum number of moves as specified in the problem statement.

Sample Input

```
3
1 1 1
2 2 2
3
1 2 3
3 2 1
4
1 1 1 1
1 1 1 1
0
```

Sample Output

```
Case 1: 7
Case 2: 3
Case 3: 0
```