



F

Few Teams will Solve This



You have just finished a compiler design homework question where you had to find the parse tree of an expression. Unfortunately you left your assignment in the library, but luckily your friend picked it up for you. Instead of e-mailing you the parse tree so that you can rewrite the solution, your friend decides to play a practical joke and sends you just the DFS and BFS trace. Rather than trying to redo the entire question you decide to reconstruct the tree. But doing that you realize there could be more than one tree that produces the given DFS and BFS trace.

Can you find out how many trees produce the given DFS and BFS trace?

Input

The input file contains several test cases as described below.

The first line of the input is the number n ($1 \leq n \leq 1000$) of nodes in the tree. The nodes in the tree are numbered 1, 2, ..., n . The remaining numbers are the BFS traversal followed by the DFS traversal. Note that when a parent was expanded the children were traversed in ascending order. The last line of input is a case where $n = 0$ and that doesn't need to be processed.

Output

For each case, output the case number first followed by the required result. Since the result could be very big, output the result modulo 19821202.

Sample Input	Sample Output
8 4 3 5 1 2 8 7 6 4 3 1 7 2 6 5 8 3 1 2 3 1 2 3 0	Case 1: 1 Case 2: 2

Problem setter: Rujia Liu Special Thanks: Renat Mulla Khanov/Sohel Hafiz