Problem F. First Solved, Last Coded

Time limit: 2 seconds
Memory limit: 1024 megabytes

In ICPC, teamwork is everything. That's why everyone on your team has a well-defined role: Sol the Solver can solve any problem in the problem set, Codie the Coder can implement any solution that Sol comes up with, and you... are the glue that holds everything together. Sol and Codie are very picky about the order of problems they would solve/code, and your job is to satisfy their preferences.

There will be n problems in the upcoming contest, and you know the general topic of each problem: greedy, geometry, graphs, etc. For simplicity, we will represent each topic with an integer from 1 to n. These integers don't have to be distinct, that is, multiple problems in the contest can have the same topic.

Sol wants to solve problems in a specific order of topics: first, the problem with the topic a_1, after that, the problem with the topic a_2, ..., and finally, the problem with the topic a_n. Codie also has a preference list: $b_1, b_2, \ldots, b_n$, only willing to code problems in that order of topics.

Your job during the contest will be to take solution sheets from Sol and hand them to Codie in the correct order. As your team only has one table to work with, you don't have enough space to arrange all the solutions neatly. Thus, you came up with the following workflow: you will ask Sol for solutions (who will hand them to you in order $a_1, a_2, \ldots, a_n$), store them in a stack on your part of the table, and hand them to Codie to code (in order $b_1, b_2, \ldots, b_n$).

More formally, at any moment during the contest, you have (at most) two actions you can make:

- If there are still any unsolved problems remaining, ask Sol for another solution and put it on top of your stack of solution sheets. This action is denoted by the character '`S`'.
- If your stack is not empty, take the solution sheet from the top of your stack and give it to Codie to implement. This action is denoted by the character '`C`'.

For the given lists of Sol's and Codie's preferences, find a sequence of actions that ensures that all problems are solved and coded in the correct order. Consider all solving and coding times insignificant — managing solution sheets is a much harder and more important job anyway.

Input

The first line contains a single integer n, denoting the number of problems in the contest ($1 \le n \le 100$).

The second line contains n integers $a_1, a_2, \ldots, a_n$, denoting Sol's preferred order of topics ($1 \le a_i \le n$).

The third line contains n integers $b_1, b_2, \ldots, b_n$, denoting Codie's preferred order of topics ($1 \le b_i \le n$).

The given lists are equal as multisets: every integer occurs the same number of times in A and in B.

Output

If your task is impossible, print "`NO`". Otherwise, print "`YES`" on the first line, followed by the sequence of actions on the second line: a string consisting of $2n$ characters '`S`' or '`C`' (n of each), describing your actions in order.

You are not allowed to ask Sol for more solutions if all n problems have already been solved, or give Codie a solution with the wrong topic. If there are multiple answers, print any of them.

Examples

standard input	standard output
4 4 1 2 2 1 2 4 2	YES SSCSCCSC
3 2 3 1 1 2 3	NO