

Kiwileaks

Input File: *leakin.txt*

Output File: *leakout.txt*

Time and Memory Limits: 1 second, 32 MB

The infamous New Zealand hacker Judy Egnassa has been causing a lot of trouble for the United Bakers of Australia. The massive multinational bakery has dominated the world’s cake production for decades and kept its recipes top-secret. However, an anonymous baker of the UBA recently released hundreds of thousands of secret recipes to Judy, who has made the information available to the whole world. This has caused an international uproar, as key secret ingredients included toenail clippings, droppings of newts, and worm eggs to make the consumer hungry for more delicious cake.

Amidst apology tours and questionable legal action taken against Judy, UBA has been working hard to find a new way to encrypt their recipes so that such an incident will never happen again. World-renowned mathematicians employed by UBA have decided that a *substitution cipher* is the most effective method of encryption. A recipe contains a series of symbols, represented by positive integers. In a substitution cipher, every symbol has a corresponding unique “substitute symbol”. Each symbol in the original recipe is replaced with its substitute symbol to produce the encrypted recipe.

In order to make the contents of the secret recipes even more secure, the mathematicians will re-encrypt each recipe multiple times using the same substitution cipher. In experimenting with this ground-breaking encryption method, the mathematicians have found that sometimes a recipe would be re-encrypted so many times that the multi-encrypted recipe would be the same as its original, unencrypted form.

For example, a recipe could be “1 2 2 5”, and a possible substitution cipher:

$$\begin{aligned} 1 &\rightarrow 5 \\ 2 &\rightarrow 3 \\ 3 &\rightarrow 2 \\ 4 &\rightarrow 1 \\ 5 &\rightarrow 4 \end{aligned}$$

After one encryption, the recipe would read “5 3 3 4”. After a second encryption, “4 2 2 1”. The first six encryptions would be:

$$1\ 2\ 2\ 5 \rightarrow 5\ 3\ 3\ 4 \rightarrow 4\ 2\ 2\ 1 \rightarrow 1\ 3\ 3\ 5 \rightarrow 5\ 2\ 2\ 4 \rightarrow 4\ 3\ 3\ 1 \rightarrow 1\ 2\ 2\ 5$$

After 6 encryptions, the multi-encrypted recipe is the same as its original unencrypted form.

The President of the UBA has determined that this could be a problem. While the ultra-secure multiple encryption method works well most of the time, choosing the number of re-encryptions poorly could result in no encryption at all, allowing Judy to easily release to the world any recipes she finds.

In order to make sure that this never happens, your task is to write a program determining the minimum number of times a recipe must be encrypted until it reaches its original unencrypted form, given the unencrypted recipe and a substitution cipher.

Input

- The first line of the input file will contain two integers N and K . There are N unique symbols in the substitution cipher and K symbols in the recipe to be encrypted ($1 \leq N, K \leq 200$). Each symbol will be represented by an integer from 1 to N .
- The second line will contain N integers. The i th integer is the substituted symbol for symbol i . Note that it is possible for the i th integer to be the integer i (i.e. a symbol may encrypt to itself). It is guaranteed that each integer from 1 to N will appear exactly once on the second line.
- The third line will contain K integers, representing the recipe to be encrypted.

Output

Your program should write one line of output containing one integer: the minimum number of times the recipe must be encrypted with the substitution cipher before it returns to its unencrypted form. If this will never happen no matter how many times the recipe is re-encrypted, output the word “Never”. It is guaranteed that if such a number does exist, it will be less than 2^{63} .

Sample Input

```
5 4
5 3 2 1 4
1 2 2 5
```

Sample Output

```
6
```

Explanation

The sample data corresponds to the example case described in the problem.

Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.

For 30% of the available marks, the correct answer will be a number less than 10 000.