

Problem C: Shuffling

Source: `shuffling.{c, cpp, java}`

Input: `console {stdin, cin, System.in}`

Output: `console {stdout, cout, System.out}`

A casino owns an expensive card shuffling machine which may shuffle up to 520 cards at a time (there are 52 cards in each deck). For convenience, we will simply label the cards 1, 2, 3, ..., N where N is the total number of cards, and copies of the same card (e.g. Ace of Spades) from different decks are considered different. Unfortunately, the card shuffling machine is defective, and it always shuffles the cards the same way. The company that produces these machines is out of business because of the economic downturn. There is no one who can fix the machine, and a new machine is too expensive.

Being a brilliant employee of the casino, you realized that all is not lost. You can shuffle the cards differently simply by using the machine zero or more times. For example, suppose that the machine shuffles the cards 1, 2, 3, 4 into the order 2, 3, 4, 1. If you put the cards into the machine, take the shuffled cards out and insert them into the machine again (without changing the order), you will get the order 3, 4, 1, 2. That way, it is possible to shuffle the cards in many different ways even though it may take longer. But this is not a significant issue since decks do not have to be reshuffled often, and used decks can be shuffled while other decks are being used to avoid any waiting time.

Unfortunately, not all shufflings can be produced in this way in general, and you wish to know if this procedure "stack the decks" in a favorable way for the casino or the player. As a first step, you wish to know which shufflings are possible to produce, and how many times you need to use the machine on the deck in order to produce the shuffling.

Input

The input for each case consists of three lines. The first line consists of a single integer N indicating the number of cards to shuffle. The number of cards is a positive integer up to 520. The second line consists of the integers 1, 2, ..., N listed in some order and separated by a space. The list gives the order of the shuffling performed by the machine when the input cards are ordered 1, 2, ..., N . The third line is in the same format as the second line, and gives the shuffling we wish to obtain. The end of input is indicated by a line in which $N = 0$.

Output

For each case, print the smallest number of times (zero or more) you need to pass the deck through the machine to produce the desired shuffling. If it is not possible, print -1. The output for each case should be in a single line. You may assume that the answer will always fit in a 32-bit signed integer.

Sample Input

```
4
2 3 4 1
3 4 1 2
4
2 3 4 1
1 3 2 4
10
2 1 3 5 6 7 8 9 10 4
1 2 3 9 10 4 5 6 7 8
0
```

Sample Output

```
2
-1
12
```