

The 1999 British Informatics Olympiad Final

Card shuffling

I have a pack of six cards, all clubs - A23456. I split the pack into two halves, A23 and 456, then interleave them with the 4 on top; in other words, I do a single in-riffle shuffle changing the order to 4A5263. Using the same shuffle a second time gives 246A35. A third identical shuffle returns the pack to its original order.

Given a pack of n cards and a method of shuffling them, we are interested in how many times this shuffle must be repeated before the pack is restored to its original order. The in-riffle shuffle on six cards restores the original order with 3 shuffles; it requires 52 shuffles for a pack of 52 cards. The corresponding out-riffle shuffle, which interleaves placing the top card of the other pile on top, takes 4 shuffles with 6 cards and 8 shuffles with 52 cards.

Write a program that first inputs an integer n ($1 \leq n \leq 100$), followed by n lines each containing a single integer. These n lines describe a shuffle; if the k th of these lines is l it indicates the card in position k moves to position l after the shuffle. You should output the number of times this shuffle must be repeated for the original order to be restored.

All the input shuffles will be well-formed. In other words, the values on the n lines will be an arrangement of the numbers $1, \dots, n$ without omission or repetition. No shuffle on up to 100 cards requires more than 2^{28} repetitions.

Sample Input

```
6
2
4
6
1
3
5
```

Sample Output

```
3
```