

Shuffles

The most common technique for shuffling a deck of cards is called the Riffle or Dovetail shuffle. The deck is split into two stacks, which are then interleaved with each other. The deck can be split anywhere, and the two stacks can be interleaved in any way.

For example, consider a deck with 10 unique cards:

1 2 3 4 5 6 7 8 9 10

Split them somewhere:

1 2 3 4 5 6 7 8 9 10

And interleave them in some way:

1 2 7 3 8 9 4 5 10 6

Do it again. Split them somewhere:

1 2 7 3 8 9 4 5 10 6

And interleave them in some way:

3 8 1 9 4 5 2 7 10 6

This is one possible ordering after 2 shuffles. Suppose there are n unique cards, and that they start out perfectly ordered: 1, 2, 3, ..., n . Given an ordering of the deck, what is the smallest number of shuffles that could possibly put the deck in that order?

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each test case will begin with a single integer n ($1 \leq n \leq 1,000,000$) indicating the number of cards in the deck. On the next line will be n unique integers c ($1 \leq c \leq n$), with a single space between them, indicating an ordering of the n cards. The values c are guaranteed to be a permutation of the numbers $1..n$.

Output

Output a single line with a single integer indicating the minimum number of shuffles that could possibly put the deck in the given order. Output no spaces.

Example

Input:

10
1 2 7 3 8 9 4 5 10 6

Output:

1

Input:

8

2 1 4 3 6 5 8 7

Output:

3

Input:

10

3 8 1 9 4 5 2 7 10 6

Output:

2