INOI, Iranian National Olympiad in Informatics
Online Contest, April 2008
First Exam

Two-stage Sorting

Time limit for each test: 2000 miliseconds
Memory limit: 200 megabytes

A parking for n cars was built in a business center. Shortly thereafter, it was found out that the small exit causes huge traffic at the end of each day. The board of directors decided, in order to solve this problem, to change the ordering of cars in the lots, in such a way that any employee who finishes earlier, should be able to leave the parking earlier as well. We know that no two employees finish their work at the same time. Parking lots form a line. The car in lot 1 is closest to the exit, while the farthest car is located in lot n.

We have a permutation π for numbers 1 through n which indicates the owner of the car in lot i is the π_i^{th} one to finish her job. In other words, there are $\pi_i - 1$ persons who finish earlier than she does. We want to reorder the cars so that the car currently at lot i should be placed at lot π_i.

A contractor company is hired to do the replacement of cars. The company has n drivers to do the job. Each is assigned to some cars. While a car is assigned to at most one driver, the number of cars (possibly zero) assigned to each driver need not be the same. They work in parallel, but each driver can only move his cars within their positions. Note that a driver cannot replace a car of his, with a car assigned to some other guy. The time taken by a driver to do his job is equal to the number of cars he is given (in minutes). Since the work is done in parallel, the total time is exactly the maximum number of cars assigned to one driver.

A board member suggested they could employ another driver to lower the total time. The new driver is to replace two cars in each step. He starts his work before the contractor. Any replacement takes one minute for this driver.

In other words, we can perform k replacement operations for the cars. Then, we need to partition the permutation into some subsets. If the largest set has l cars, the total time taken to do the reordering is $l + k$ minutes.

For example, let $\pi = \langle 4, 3, 1, 6, 5, 2 \rangle$. We can first change the position of first and last car (in one minute) to get the permutation $\langle 2, 3, 1, 6, 5, 4 \rangle$. Then first, second and third cars are given to one driver, whereas the fourth and sixth are given to another. The fifth car is not assigned at all. Clearly, we can reorder each set to obtain $\langle 1, 2, 3, 4, 5, 6 \rangle$. This takes three minutes. Therefore, we need four minutes to perform the whole process.

Problem

Write a program that

- Reads a permutation π from the *Standard Input*.
- Computes the time needed to reorder the cars.
- Writes the result to the *Standard Output*.

Input Sepcification

The first line of input contains one integer n.
The next line contains n numbers, separated with one space, denoting a permutation from 1 through n.

Output Specification

In one line, write the minimum amount of time, in minutes, to do the reodering of cars, given by the above permutation, using the above-mentioned operations.

Restrictions

- $1 \leq n \leq 1,000,000$.
- In 30% of the test cases, n will not exceed $20,000$.

Example

Standard Input	Standard Output
6 4 3 1 6 5 2	4