Problem C
Improved Random Sort

Input file: *testdata.in*
Time limit: 1 second

Problem Description

A scientist Dinger invented "improved random sort", and we would like to use his new algorithm to sort a list of N distinct integers into increasing order.

The sort is done in N iterations. At the beginning of the i-th iteration the first $i-1$ keys are already in increasing order, so we need to check if the first i numbers are in increasing sorted order as well. If they are in increasing order, then we are done. If the first i numbers are *not* in increasing order, then we repeatedly call a random function $rand()$ to get a random integer r between $[1, i)$, until the first i numbers are in increasing order if we inserted $n[i]$ to the left of $n[r]$. After N iterations the keys will be in sorted order.

Obviously we may call the random function more than once during an iteration. Despite that we do not know exactly how many times the random function will be called, we can calculate its expected value, which is denoted by E. Now for a list of integers please calculate E, the expected number of times the random function will be called.

Technical Specification

- $1 \leq N \leq 10^5$
- $1 \leq n_i \leq 10^6$

Input Format

The input have multiple test cases. Each test case contains two lines – the integer N in the first line, and the list of N distinct integers n_i $(1 \leq i \leq N)$ separated by white spaces in the second line. There are no blank lines between test cases.

Output Format

For each test case output the expected value of E in a line to 3 decimal places.

Sample Input

```
3
1 2 3
3
3 1 2
```

Sample Output

```
0.000
3.000
```