Lazy Sorting

HackerRank

Logan is cleaning his apartment. In particular, he must sort his old favorite sequence, P, of N positive integers in nondecreasing order. He's tired from a long day, so he invented an easy way (in his opinion) to do this job. His algorithm can be described by the following pseudocode:

```
while isNotSorted(P) do {
    WaitOneMinute();
    RandomShuffle(P)
}
```

Can you determine the expected number of minutes that Logan will spend waiting for P to be sorted?

Input Format

The first line contains a single integer, N, denoting the size of permutation P.
The second line contains N space-separated integers describing the respective elements in the sequence's current order, $P_0, P_1, \ldots, P_{N-1}$.

Constraints

- $2 \leq N \leq 18$
- $1 \leq P_i \leq 100$

Output Format

Print the expected number of minutes Logan must wait for P to be sorted, correct to 6 decimal places.

Sample Input

```
2
5 2
```

Sample Output

```
2.000000
```

Explanation

There are two permutations possible after a random shuffle, and each of them has probability 0.5. The probability to get the sequence sorted after the first minute is 0.5. The probability that P will be sorted after the second minute is 0.25, the probability P will be sorted after the third minute is 0.125, and so on. So, the answer is equal to the following sum:

$$\sum_{i=1}^{\infty} i \times 2^{-i} = 2$$