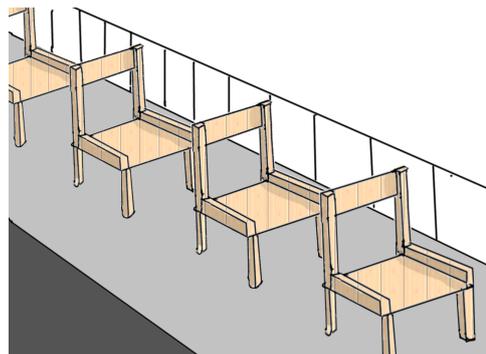


Boarding Passes (passes)

After successfully navigating the local traditions, you managed to catch the ferry just in time for its departure. However, you did not expect that many people to be headed for Lübeck! Since you do not want to be late for the award ceremony,* you want to speed up the boarding process of the ferry.

The ferry has a single row of N seats which is fully booked by N passengers. Each passenger has a ticket that gives their assigned seat and one out of G boarding groups.

When boarding, these groups are called one after the other. The passengers within each boarding group will board in a random order, with all possible orders being equally likely. Each passenger can board the ferry either at the front or back of the row of seats and will then move to their assigned seat before another passenger boards.



You determined that the most time-consuming element of this process is when a passenger has to *pass* someone who is already seated.† Fortunately you found a staff uniform in a nearby locker, so you can decide the order in which the groups are boarding and tell each passenger *before the boarding begins* whether to board the ferry from the front or back of the row of seats.

Write a program that uses the ticket information to calculate the minimal expected number of passes during boarding if you choose the order of the boarding groups and the assignment of the passengers to the front and back optimally.

Note

Given an order of the boarding groups and an assignment of the passengers to the front and back, the expected number of passes is defined as

$$1 \cdot p_1 + 2 \cdot p_2 + 3 \cdot p_3 + \dots$$

where p_k is the probability that there are exactly k passes during boarding. Put differently, this is the average number of passes among all possible orders of the passengers in each boarding group.

Input

The input consists of a string of N characters $s_1 \dots s_N$ where s_i is one of the first G uppercase characters of the English alphabet, representing the boarding group of the passenger assigned to the i -th seat in the row (with the seat at the front of the row being seat 1).

Output

Your program should output a single number, the minimal expected number of passes if you choose the order of the boarding groups and the assignment of the passengers to the front and back optimally. Your answer will be accepted if its absolute error is at most 0.001.

* You will also need some time to store all your stolen art in the hostel.

† The luggage containing all those ties is a considerable obstacle in the aisle.



Constraints

We always have $1 \leq G \leq 15$ and $1 \leq N \leq 100\,000$.

Subtask 1 (5 points). $G = 1$, i.e. there is only one boarding group.

Subtask 2 (25 points). $G \leq 7$ and $N \leq 100$

Subtask 3 (30 points). $G \leq 10$ and $N \leq 10\,000$

Subtask 4 (40 points). No further constraints.

Examples

Input	Output
AACCAA	1
HEHEHEHILOL	7.5
ONMLKJIHGFE DCBAABCDEFGHIJKLMNO	0
AAAAAAAAAAAAA . . . AAAAAAAAAAAAAA	100800.5

In the fourth example, the input consists of the character A repeated 899 times. All example inputs can be found in the attachment for this task in CMS.

In the first example, group C should board before group A and the passengers in the 1st, 2nd, and 3rd seats should board from the front while the rest should board from the back.

This makes it impossible for the two passengers of group C to pass each other, and they will not pass any passenger of group A since group C boards first. Moreover, the passengers of group A will not pass any passenger of group C since all passengers of group A boarding from the front are seated in front of the passengers of group C and all passengers of group A boarding from the back are seated behind the passengers of group C.

Therefore, the only possible passes are the passenger in the 2nd seat passing the passenger in the 1st seat, which will only happen if the passenger in the 1st seat boards before the passenger in the 2nd seat, and the same for the passengers in the 5th and 6th seats. Since each of these events occurs with a probability of 50%, it follows that the expected number of passes is equal to 1.

Limits

Time: 2 s

Memory: 1024 MiB