Codeforces Round #373 (Div. 1)

A. Efim and Strange Grade

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Efim just received his grade for the last test. He studies in a special school and his grade can be equal to any positive decimal fraction. First he got disappointed, as he expected a way more pleasant result. Then, he developed a tricky plan. Each second, he can ask his teacher to round the grade at any place after the decimal point (also, he can ask to round to the nearest integer).

There are t seconds left till the end of the break, so Efim has to act fast. Help him find what is the maximum grade he can get in no more than t seconds. Note, that he can choose to not use all t seconds. Moreover, he can even choose to not round the grade at all.

In this problem, classic rounding rules are used: while rounding number to the n-th digit one has to take a look at the digit $n+1$. If it is less than 5 than the n-th digit remain unchanged while all subsequent digits are replaced with 0. Otherwise, if the $n+1$ digit is greater or equal to 5, the digit at the position n is increased by 1 (this might also change some other digits, if this one was equal to 9) and all subsequent digits are replaced with 0. At the end, all trailing zeroes are thrown away.

For example, if the number 1.14 is rounded to the first decimal place, the result is 1.1, while if we round 1.5 to the nearest integer, the result is 2. Rounding number 1.299996121 in the fifth decimal place will result in number 1.3.

Input

The first line of the input contains two integers n and t ($1 \le n \le 200\,000$, $1 \le t \le 10^9$) — the length of Efim's grade and the number of seconds till the end of the break respectively.

The second line contains the grade itself. It's guaranteed that the grade is a positive number, containing at least one digit after the decimal points, and it's representation doesn't finish with 0.

Output

Print the maximum grade that Efim can get in t seconds. Do not print trailing zeroes.

Examples

input

```
6 1
10.245
```

output

```
10.25
```

input

```
6 2
10.245
```

output

```
10.3
```

input

```
3 100
9.2
```

output

```
9.2
```

Note

In the first two samples Efim initially has grade 10.245.

During the first second Efim can obtain grade 10.25, and then 10.3 during the next second. Note, that the answer 10.30 will be considered incorrect.

In the third sample the optimal strategy is to not perform any rounding at all.

C. Sasha and Array

time limit per test: 5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Sasha has an array of integers $a_1, a_2, \ldots, a_n$. You have to perform m queries. There might be queries of two types:

1. `1 l r x` — increase all integers on the segment from l to r by values x;
2. `2 l r` — find $\sum_{i=l}^{r} f(a_i)$, where $f(x)$ is the x-th Fibonacci number. As this number may be large, you only have to find it modulo $10^9 + 7$.

In this problem we define Fibonacci numbers as follows: $f(1) = 1, f(2) = 1, f(x) = f(x - 1) + f(x - 2)$ for all $x > 2$.

Sasha is a very talented boy and he managed to perform all queries in five seconds. Will you be able to write the program that performs as well as Sasha?

Input

The first line of the input contains two integers n and m $(1 \le n \le 100\,000, 1 \le m \le 100\,000)$ — the number of elements in the array and the number of queries respectively.

The next line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$.

Then follow m lines with queries descriptions. Each of them contains integers tp_i, l_i, r_i and may be x_i $(1 \le tp_i \le 2, 1 \le l_i \le r_i \le n, 1 \le x_i \le 10^9)$. Here $tp_i = 1$ corresponds to the queries of the first type and tp_i corresponds to the queries of the second type.

It's guaranteed that the input will contains at least one query of the second type.

Output

For each query of the second type print the answer modulo $10^9 + 7$.

Examples

input

```
5 4
1 1 2 1 1
2 1 5
```

```
1 2 4 2
2 2 4
2 1 5
```

output

```
5
7
9
```

Note

Initially, array a is equal to $1, 1, 2, 1, 1$.

The answer for the first query of the second type is $f(1) + f(1) + f(2) + f(1) + f(1) = 1 + 1 + 1 + 1 + 1 = 5$.

After the query `1 2 4 2` array a is equal to $1, 3, 4, 3, 1$.

The answer for the second query of the second type is $f(3) + f(4) + f(3) = 2 + 3 + 2 = 7$.

The answer for the third query of the second type is $f(1) + f(3) + f(4) + f(3) + f(1) = 1 + 2 + 3 + 2 + 1 = 9$.

D. Andrew and Chemistry

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

During the chemistry lesson Andrew learned that the saturated hydrocarbons (alkanes) enter into radical chlorination reaction. Andrew is a very curious boy, so he wondered how many different products of the reaction may be forms for a given alkane. He managed to solve the task for small molecules, but for large ones he faced some difficulties and asks you to help.

Formally, you are given a tree consisting of n vertices, such that the degree of each vertex doesn't exceed 4. You have to count the number of distinct non-isomorphic trees that can be obtained by adding to this tree one new vertex and one new edge, such that the graph is still the tree and the degree of each vertex doesn't exceed 4.

Two trees are isomorphic if there exists a bijection $f(v)$ such that vertices u and v are connected by an edge if and only if vertices $f(v)$ and $f(u)$ are connected by an edge.

Input

The first line of the input contains an integer n $(1 \le n \le 100\ 000)$ — the number of vertices in the tree.

Then follow $n - 1$ lines with edges descriptions. Each edge is given by two integers u_i and v_i $(1 \le u_i, v_i \le n)$ — indices of vertices connected by an edge. It's guaranteed that the given graph is a tree and the degree of each vertex doesn't exceed 4.

Output

Print one integer — the answer to the question.

Examples

input

```
4
1 2
2 3
2 4
```

output

```
2
```

input

```
5
1 2
1 3
1 4
1 5
```

output

```
1
```

input

```
5
2 5
5 3
4 3
4 1
```

output

```
3
```

Note

In the first sample, one can add new vertex to any existing vertex, but the trees we obtain by adding a new vertex to vertices 1, 3 and 4 are isomorphic, thus the answer is 2.

In the second sample, one can't add new vertex to the first vertex, as its degree is already equal to four. Trees, obtained by adding a new vertex to vertices 2, 3, 4 and 5 are isomorphic, thus the answer is 1.

E. Matvey's Birthday

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Today is Matvey's birthday. He never knows what to ask as a present so friends gave him a string s of length n. This string consists of only first eight English letters: 'a', 'b', …, 'h'.

First question that comes to mind is: who might ever need some string? Matvey is a special boy so he instantly found what to do with this string. He used it to build an undirected graph where vertices correspond to position in the string and there is an edge between distinct positions a and b $(1 \le a, b \le n)$ if **at least one** of the following conditions hold:

1. a and b are neighbouring, i.e. $|a - b| = 1$.
2. Positions a and b contain equal characters, i.e. $s_a = s_b$.

Then Matvey decided to find the diameter of this graph. Diameter is a maximum distance (length of the shortest path) among all pairs of vertices. Also, Matvey wants to find the number of pairs of vertices such that the distance between them is equal to the diameter of the graph. As he is very cool and experienced programmer he managed to solve this problem very fast. Will you do the same?

Input

The first line of the input contains a single integer n $(2 \le n \le 100\,000)$ — the length of the string.

The second line contains the string s itself. It's guaranteed that s consists of only first eight letters of English alphabet.

Output

Print two integers — the diameter of the graph and the number of pairs of positions with the distance equal to the diameter.

Examples

input

```
3
abc
```

output

```
2 1
```

input

```
7
aaabaaa
```

output

```
2 4
```

Note

Consider the second sample.

The maximum distance is 2. It's obtained for pairs $(1, 4)$, $(2, 4)$, $(4, 6)$ and $(4, 7)$.