A. Kyoya and Photobooks

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Kyoya Ootori is selling photobooks of the Ouran High School Host Club. He has 26 photos, labeled "a" to "z", and he has compiled them into a photo booklet with some photos in some order (possibly with some photos being duplicated). A photo booklet can be described as a string of lowercase letters, consisting of the photos in the booklet in order. He now wants to sell some "special edition" photobooks, each with one extra photo inserted anywhere in the book. He wants to make as many distinct photobooks as possible, so he can make more money. He asks Haruhi, how many distinct photobooks can he make by inserting one extra photo into the photobook he already has?

Please help Haruhi solve this problem.

Input

The first line of input will be a single string s ($1 \le |s| \le 20$). String s consists only of lowercase English letters.

Output

Output a single integer equal to the number of distinct photobooks Kyoya Ootori can make.

Examples

input

```
a
```

output

```
51
```

input

```
hi
```

output

```
76
```

Note

In the first case, we can make 'ab','ac',...,'az','ba','ca',...,'za', and 'aa', producing a total of 51 distinct photo booklets.

B. Ohana Cleans Up

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ohana Matsumae is trying to clean a room, which is divided up into an n by n grid of squares. Each square is initially either clean or dirty. Ohana can sweep her broom over columns of the grid. Her broom is very strange: if she sweeps over a clean square, it will become dirty, and if she sweeps over a dirty square, it will become clean. She wants to sweep some columns of the room to maximize the number of rows that are completely clean. It is not allowed to sweep over the part of the column, Ohana can only sweep the whole column.

Return the maximum number of rows that she can make completely clean.

Input

The first line of input will be a single integer n ($1 \le n \le 100$).

The next n lines will describe the state of the room. The i-th line will contain a binary string with n characters denoting the state of the i-th row of the room. The j-th character on this line is '1' if the j-th square in the i-th row is clean, and '0' if it is dirty.

Output

The output should be a single line containing an integer equal to a maximum possible number of rows that are completely clean.

Examples

input

```
4
0101
1000
1111
0101
```

output

```
2
```

input

```
3
111
111
111
```

output

```
3
```

Note

In the first sample, Ohana can sweep the 1st and 3rd columns. This will make the 1st and 4th row be completely clean.

In the second sample, everything is already clean, so Ohana doesn't need to do anything.

C. Kyoya and Colored Balls

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Kyoya Ootori has a bag with n colored balls that are colored with k different colors. The colors are labeled from 1 to k. Balls of the same color are indistinguishable. He draws balls from the bag one by one until the bag is empty. He noticed that he drew the last ball of color i before drawing the last ball of color $i + 1$ for all i from 1 to $k - 1$. Now he wonders how many different ways this can happen.

Input

The first line of input will have one integer k ($1 \leq k \leq 1000$) the number of colors.

Then, k lines will follow. The i-th line will contain c_i, the number of balls of the i-th color ($1 \leq c_i \leq 1000$).

The total number of balls doesn't exceed 1000.

Output

A single integer, the number of ways that Kyoya can draw the balls from the bag as described in the statement, modulo $1\,000\,000\,007$.

Examples

input

```
3
2
2
1
```

output

```
3
```

input

```
4
1
2
3
4
```

output

```
1680
```

Note

In the first sample, we have 2 balls of color 1, 2 balls of color 2, and 1 ball of color 3. The three ways for Kyoya are:

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

D. Kyoya and Permutation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let's define the permutation of length n as an array $p = [p_1, p_2, ..., p_n]$ consisting of n distinct integers from range from 1 to n. We say that this permutation maps value 1 into the value p_1, value 2 into the value p_2 and so on.

Kyota Ootori has just learned about *cyclic representation* of a permutation. A *cycle* is a sequence of numbers such that each element of this sequence is being mapped into the next element of this sequence (and the last element of the cycle is being mapped into the first element of the cycle). The *cyclic representation* is a representation of p as a collection of cycles forming p. For example, permutation $p = [4, 1, 6, 2, 5, 3]$ has a *cyclic representation* that looks like $(142)(36)(5)$ because 1 is replaced by 4, 4 is replaced by 2, 2 is replaced by 1, 3 and 6 are swapped, and 5 remains in place.

Permutation may have several cyclic representations, so Kyoya defines the *standard cyclic representation* of a permutation as follows. First, reorder the elements within each cycle so the largest element is first. Then, reorder all of the cycles so they are sorted by their first element. For our example above, the *standard cyclic representation* of $[4, 1, 6, 2, 5, 3]$ is $(421)(5)(63)$.

Now, Kyoya notices that if we drop the parenthesis in the standard cyclic representation, we get another permutation! For instance, $[4, 1, 6, 2, 5, 3]$ will become $[4, 2, 1, 5, 6, 3]$.

Kyoya notices that some permutations don't change after applying operation described above at all. He wrote all permutations of length n that do not change in a list in lexicographic order. Unfortunately, his friend Tamaki Suoh lost this list. Kyoya wishes to reproduce the list and he needs your help. Given the integers n and k, print the permutation that was k-th on Kyoya's list.

Input

The first line will contain two integers n, k ($1 \le n \le 50$, $1 \le k \le min\{10^{18}, l\}$ where l is the length of the Kyoya's list).

Output

Print n space-separated integers, representing the permutation that is the answer for the question.

Examples

input
```
4 3
```
output
```
1 3 2 4
```

input
```
10 1
```
output
```
1 2 3 4 5 6 7 8 9 10
```

Note

The standard cycle representation is $(1)(32)(4)$, which after removing parenthesis gives us the original permutation. The first permutation on the list would be $[1, 2, 3, 4]$, while the second permutation would be $[1, 2, 4, 3]$.

E. Love Triangles

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are many anime that are about "love triangles": Alice loves Bob, and Charlie loves Bob as well, but Alice hates Charlie. You are thinking about an anime which has n characters. The characters are labeled from 1 to n. Every pair of two characters can either mutually love each other or mutually hate each other (there is no neutral state).

You hate love triangles (A-B are in love and B-C are in love, but A-C hate each other), and you also hate it when nobody is in love. So, considering any three characters, you will be happy if exactly one pair is in love (A and B love each other, and C hates both A and B), or if all three pairs are in love (A loves B, B loves C, C loves A).

You are given a list of m known relationships in the anime. You know for sure that certain pairs love each other, and certain pairs hate each other. You're wondering how many ways you can fill in the remaining relationships so you are happy with every triangle. Two ways are considered different if two characters are in love in one way but hate each other in the other. Print this count modulo $1\,000\,000\,007$.

Input

The first line of input will contain two integers n, m ($3 \le n \le 100\,000$, $0 \le m \le 100\,000$).

The next m lines will contain the description of the known relationships. The i-th line will contain three integers a_i, b_i, c_i. If c_i is 1, then a_i and b_i are in love, otherwise, they hate each other ($1 \le a_i, b_i \le n$, $a_i \ne b_i$, $c_i \in \{0, 1\}$).

Each pair of people will be described no more than once.

Output

Print a single integer equal to the number of ways to fill in the remaining pairs so that you are happy with every triangle modulo $1\,000\,000\,007$.

Examples

input

```
3 0
```

output

```
4
```

input

```
4 4
1 2 1
2 3 1
3 4 0
4 1 0
```

output

```
1
```

input

```
4 4
1 2 1
2 3 1
3 4 0
4 1 1
```

output

```
0
```

Note

In the first sample, the four ways are to:

- Make everyone love each other
- Make 1 and 2 love each other, and 3 hate 1 and 2 (symmetrically, we get 3 ways from this).

In the second sample, the only possible solution is to make 1 and 3 love each other and 2 and 4 hate each other.