A. Bad Encryption

1 second, 256 megabytes

Given a string s consisted of lower-case letters, we encrypt it by the following two steps:

1. Merge all consecutive same letters to a single letter. For example, `aaaaaaabbbbbbbcc` will be replaced by `abc` after the first step.
2. Insert a pair of same lower-case letters to the string. Repeat the second step for multiple times.

Here is an instance of the encryption algorithm. Initially the string s is `ssstieerrrliiittz`.

After the first step, the string will be `stierlitz`.

After inserting `dd`, the string will be `stddierlitz`.

After inserting `aa`, the string will be `stdaadierlitz`.

After inserting `ww`, the string will be `wwstdaadierlitz`. The algorithm returns `wwstdaadierlitz`.

Given a string returned by the encryption algorithm. Output the string obtained after invoking the first step.

Input

A single string which denotes the string returned by the encryption algorithm. The length of the string is at most 1000000.

Output

Display the string obtained after invoking the first step.

input
wwstdaadierlitz
output
stierlitz

Problem B
Rainbow Strings
Time limit: 1 second

Define a string to be a *rainbow string* if every letter in the string is distinct. An empty string is also considered a *rainbow string*.

Given a string of lowercase letters, compute the number of different subsequences which are *rainbow strings*. Two subsequences are different if an index is included in one subsequence but not the other, even if the resulting strings are identical.

In the first example, there are 8 subsequences. The only subsequences that aren't rainbow strings are `aa` and `aab`. The remaining 6 subsequences are rainbow strings.

Input

The input will consist of a single line with a single string consisting solely of lowercase letters. The length of the string is between 1 and $100\,000$ (inclusive).

Output

Write on a single line the number of rainbow sequences, modulo the prime $11\,092\,019$.

Examples

Sample Input 1	Sample Output 1
`aab`	`6`

Sample Input 2	Sample Output 2
`icpcprogrammingcontest`	`209952`

C k-close subsets

1 second, 256 megabytes

You're given a set P of n points in the plane with integer coordinates, along with an integer k. A set Q of points is called k-close if each one of the points of Q is distance at most k away from each other point of Q. Find the cardinality of the largest subset of P that is k-close.

Input

The first line contains a single integer T ($1 \leq T \leq 10$) which is the number of test cases. For each test case, the first line contains two integers n ($1 \leq n \leq 100$) and k ($0 \leq k \leq 10000$). Each of the following n lines contains two integers which are coordinates of the input points.

It is guaranteed that the input coordinate of all points are in $[-10000, 10000]$.

Output

Print a single integer, the cardinality of the largest k-close subset of P.

input

```
2
5 20
0 0
0 2
100 100
100 110
100 120
4 1
0 0
0 1
1 0
1 1
```

output

```
3
2
```

Problem D
Carry Cam Failure
Time limit: 1 second

A Cam from a Babbage Analytical Engine

“Drat!” cursed Charles. “This stupid carry bar is not working in my Engine! I just tried to calculate the square of a number, but it’s wrong; all of the carries are lost.”

“Hmm,” mused Ada, “arithmetic without carries! I wonder if I can figure out what your original input was, based on the result I see on the Engine.”

Carryless addition, denoted by $\oplus$, is the same as normal addition, except any carries are ignored (in base 10). Thus, $37 \oplus 48$ is 75, not 85.

Carryless multiplication, denoted by $\otimes$, is performed using the schoolboy algorithm for multiplication, column by column, but the intermediate additions are calculated using *carryless addition*. More formally, Let $a_m a_{m-1} \ldots a_1 a_0$ be the digits of a, where a_0 is its least significant digit. Similarly define $b_n b_{n-1} \ldots b_1 b_0$ be the digits of b. The digits of $c = a \otimes b$ are given by the following equation:

$$c_k = a_0 b_k \oplus a_1 b_{k-1} \oplus \cdots \oplus a_{k-1} b_1 \oplus a_k b_0,$$

where any a_i or b_j is considered zero if $i > m$ or $j > n$. For example, $9 \otimes 1\,234$ is $9\,876$, $90 \otimes 1\,234$ is $98\,760$, and $99 \otimes 1\,234$ is $97\,536$.

Given N, find the smallest positive integer a such that $a \otimes a = N$.

Input

The input consists of a single line with an integer N, with at most 25 digits and no leading zeros.

Output

Print, on a single line, the least positive number a such that $a \otimes a = N$. If there is no such a, print '`-1`' instead.

Examples

Sample Input 1	Sample Output 1
`6`	`4`

Sample Input 2	Sample Output 2
`149`	`17`

Sample Input 3	Sample Output 3
`123476544`	`11112`

Sample Input 4	Sample Output 4
`15`	`-1`

Problem E
Error Correction
Time limit: 1 second

You are given W, a set of N words that are anagrams of each other. There are no duplicate letters in any word. A set of words $S \subseteq W$ is called "swap-free" if there is no way to turn a word $x \in S$ into another word $y \in S$ by swapping only a single pair of (not necessarily adjacent) letters in x. Find the size of the largest swap-free set S chosen from the given set W.

Input

The first line of input contains an integer N ($1 \leq N \leq 500$). Following that are N lines each with a single word. Every word contains only lowercase English letters and no duplicate letters. All N words are unique, have at least one letter, and every word is an anagram of every other word.

Output

Output the size of the largest swap-free set.

Sample Input 1	Sample Output 1
6 abc acb cab cba bac bca	3

Sample Input 2	Sample Output 2
11 alerts alters artels estral laster ratels salter slater staler stelar talers	8

Sample Input 3	Sample Output 3
6 ates east eats etas sate teas	4

F. Coprime Factorial

1 second, 256 megabytes

Given n and m with $n \geq m$. Count the number of integers between 1 and $n!$ which is coprime with $m!$.

Output the result modulo $10^9 + 7$.

Input

Two integers n and m with $1 \leq m \leq n \leq 10^6$.

Output

Display the result modulo $10^9 + 7$.

input
4 2
output
12