A. Music

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

Little Lesha loves listening to music via his smartphone. But the smartphone doesn't have much memory, so Lesha listens to his favorite songs in a well-known social network InTalk.

Unfortunately, internet is not that fast in the city of Ekaterinozavodsk and the song takes a lot of time to download. But Lesha is quite impatient. The song's duration is T seconds. Lesha downloads the first S seconds of the song and plays it. When the playback reaches the point that has not yet been downloaded, Lesha immediately plays the song from the start (the loaded part of the song stays in his phone, and the download is continued from the same place), and it happens until the song is downloaded completely and Lesha listens to it to the end. For q seconds of real time the Internet allows you to download $q - 1$ seconds of the track.

Tell Lesha, for how many times he will start the song, including the very first start.

Input

The single line contains three integers T, S, q ($2 \le q \le 10^4$, $1 \le S < T \le 10^5$).

Output

Print a single integer — the number of times the song will be restarted.

Sample test(s)

input
```
5 2 2
```
output
```
2
```

input
```
5 4 7
```
output
```
1
```

input
```
6 2 3
```
output
```
1
```

Note

In the first test, the song is played twice faster than it is downloaded, which means that during four first seconds Lesha reaches the moment that has not been downloaded, and starts the song again. After another two seconds, the song is downloaded completely, and thus, Lesha starts the song twice.

In the second test, the song is almost downloaded, and Lesha will start it only once.

In the third sample test the download finishes and Lesha finishes listening at the same moment. Note that song isn't restarted in this case.

B. Inventory

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

Companies always have a lot of equipment, furniture and other things. All of them should be tracked. To do this, there is an inventory number assigned with each item. It is much easier to create a database by using those numbers and keep the track of everything.

During an audit, you were surprised to find out that the items are not numbered sequentially, and some items even share the same inventory number! There is an urgent need to fix it. You have chosen to make the numbers of the items sequential, starting with 1. Changing a number is quite a time-consuming process, and you would like to make maximum use of the current numbering.

You have been given information on current inventory numbers for n items in the company. Renumber items so that their inventory numbers form a *permutation* of numbers from 1 to n by changing the number of as few items as possible. Let us remind you that a set of n numbers forms a *permutation* if all the numbers are in the range from 1 to n, and no two numbers are equal.

Input

The first line contains a single integer n — the number of items ($1 \leq n \leq 10^5$).

The second line contains n numbers $a_1, a_2, ..., a_n$ ($1 \leq a_i \leq 10^5$) — the initial inventory numbers of the items.

Output

Print n numbers — the final inventory numbers of the items in the order they occur in the input. If there are multiple possible answers, you may print any of them.

Sample test(s)

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

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

input
```
1
2
```
output
```
1
```

Note

In the first test the numeration is already a permutation, so there is no need to change anything.

In the second test there are two pairs of equal numbers, in each pair you need to replace one number.

In the third test you need to replace 2 by 1, as the numbering should start from one.

C. Primes or Palindromes?

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Rikhail Mubinchik believes that the current definition of prime numbers is obsolete as they are too complex and unpredictable. A palindromic number is another matter. It is aesthetically pleasing, and it has a number of remarkable properties. Help Rikhail to convince the scientific community in this!

Let us remind you that a number is called *prime* if it is integer larger than one, and is not divisible by any positive integer other than itself and one.

Rikhail calls a number a *palindromic* if it is integer, positive, and its decimal representation without leading zeros is a palindrome, i.e. reads the same from left to right and right to left.

One problem with prime numbers is that there are too many of them. Let's introduce the following notation: $\pi(n)$ — the number of primes no larger than n, $rub(n)$ — the number of palindromic numbers no larger than n. Rikhail wants to prove that there are a lot more primes than palindromic ones.

He asked you to solve the following problem: for a given value of the coefficient A find the maximum n, such that $\pi(n) \le A \cdot rub(n)$.

Input

The input consists of two positive integers p, q, the numerator and denominator of the fraction that is the value of A ($A = \frac{p}{q}$, $p, q \le 10^4, \frac{1}{42} \le \frac{p}{q} \le 42$).

Output

If such maximum number exists, then print it. Otherwise, print "`Palindromic tree is better than splay tree`" (without the quotes).

Sample test(s)

input
```
1 1
```
output
```
40
```

input
```
1 42
```
output
```
1
```

input
```
6 4
```
output
```
172
```

D. Symmetric and Transitive

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Johnny has recently learned about set theory. Now he is studying binary relations. You've probably heard the term "equivalence relation". These relations are very important in many areas of mathematics. For example, the equality of the two numbers is an equivalence relation.

A set ρ of pairs (a, b) of elements of some set A is called a binary relation on set A. For two elements a and b of the set A we say that they are in relation ρ, if pair $(a, b) \in \rho$, in this case we use a notation $a \overset{\rho}{\sim} b$.

Binary relation is *equivalence relation*, if:

1. It is reflexive (for any a it is true that $a \overset{\rho}{\sim} a$);
2. It is symmetric (for any a, b it is true that if $a \overset{\rho}{\sim} b$, then $b \overset{\rho}{\sim} a$);
3. It is transitive (if $a \overset{\rho}{\sim} b$ and $b \overset{\rho}{\sim} c$, than $a \overset{\rho}{\sim} c$).

Little Johnny is not completely a fool and he noticed that the first condition is not necessary! Here is his "proof":

Take any two elements, a and b. If $a \overset{\rho}{\sim} b$, then $b \overset{\rho}{\sim} a$ (according to property (2)), which means $a \overset{\rho}{\sim} a$ (according to property (3)).

It's very simple, isn't it? However, you noticed that Johnny's "proof" is wrong, and decided to show him a lot of examples that prove him wrong.

Here's your task: count the number of binary relations over a set of size n such that they are symmetric, transitive, but not an equivalence relations (i.e. they are not reflexive).

Since their number may be very large (not 0, according to Little Johnny), print the remainder of integer division of this number by $10^9 + 7$.

Input

A single line contains a single integer n $(1 \le n \le 4000)$.

Output

In a single line print the answer to the problem modulo $10^9 + 7$.

Sample test(s)

input
```
1
```
output
```
1
```

input
```
2
```
output
```
3
```

input
```
3
```
output
```
10
```

Note

If $n = 1$ there is only one such relation — an empty one, i.e. $\rho = \varnothing$. In other words, for a single element x of set A the following is hold: $x \overset{\rho}{\nsim} x$.

If $n = 2$ there are three such relations. Let's assume that set A consists of two elements, x and y. Then the valid relations are $\rho = \varnothing$, $\rho = \{(x, x)\}$, $\rho = \{(y, y)\}$. It is easy to see that the three listed binary relations are symmetric and transitive relations, but they are not equivalence relations.

E. New Language

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

Living in Byteland was good enough to begin with, but the good king decided to please his subjects and to introduce a national language. He gathered the best of wise men, and sent an expedition to faraway countries, so that they would find out all about how a language should be designed.

After some time, the wise men returned from the trip even wiser. They locked up for six months in the dining room, after which they said to the king: "there are a lot of different languages, but almost all of them have letters that are divided into vowels and consonants; in a word, vowels and consonants must be combined correctly."

There are very many rules, all of them have exceptions, but our language will be deprived of such defects! We propose to introduce a set of formal rules of combining vowels and consonants, and include in the language all the words that satisfy them.

The rules of composing words are:

- The letters are divided into vowels and consonants in some certain way;
- All words have a length of exactly n;
- There are m rules of the form (pos_1, t_1, pos_2, t_2). Each rule is: if the position pos_1 has a letter of type t_1, then the position pos_2 has a letter of type t_2.

You are given some string s of length n, it is not necessarily a correct word of the new language. Among all the words of the language that lexicographically not smaller than the string s, find the minimal one in lexicographic order.

Input

The first line contains a single line consisting of letters 'V' (Vowel) and 'C' (Consonant), determining which letters are vowels and which letters are consonants. The length of this string l is the size of the alphabet of the new language ($1 \le l \le 26$). The first l letters of the English alphabet are used as the letters of the alphabet of the new language. If the i-th character of the string equals to 'V', then the corresponding letter is a vowel, otherwise it is a consonant.

The second line contains two integers n, m ($1 \le n \le 200$, $0 \le m \le 4n(n - 1)$) — the number of letters in a single word and the number of rules, correspondingly.

Next m lines describe m rules of the language in the following format: pos_1, t_1, pos_2, t_2 ($1 \le pos_1, pos_2 \le n$, $pos_1 \ne pos_2$, $t_1, t_2 \in \{$'V', 'C'$\}$).

The last line contains string s of length n, consisting of the first l small letters of the English alphabet.

It is guaranteed that no two rules are the same.

Output

Print a smallest word of a language that is lexicographically not smaller than s. If such words does not exist (for example, if the language has no words at all), print "-1" (without the quotes).

Sample test(s)

input

```
VC
2 1
1 V 2 C
aa
```

output

```
ab
```

input

```
VC
2 1
1 C 2 V
bb
```

output

```
-1
```

input

```
VCC
4 3
1 C 2 V
2 C 3 V
3 V 4 V
abac
```

output

```
acaa
```

Note

In the first test word "`aa`" is not a word of the language, but word "`ab`" is.

In the second test out of all four possibilities only word "`bb`" is not a word of a language, but all other words are lexicographically less, so there is no answer.

In the third test, due to the last rule, "`abac`" doesn't belong to the language ("`a`" is a vowel, "`c`" is a consonant). The only word with prefix "`ab`" that meets the given rules is "`abaa`". But it is less than "`abac`", so the answer will be "`acaa`"