Sponsored by Telegram

Codeforces Round #315 (Div. 1)

A. 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Rikhail Mubinchik believes that the current definition of prime numbers is obsolete as they are too complex and unpredictable.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).

Examples

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

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

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

B. 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 $10^9 + 7$.

Examples

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.

C. 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).

Examples

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"

D. Sign Posts

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

One Khanate had a lot of roads and very little wood. Riding along the roads was inconvenient, because the roads did not have road signs indicating the direction to important cities.

The Han decided that it's time to fix the issue, and ordered to put signs on every road. The Minister of Transport has to do that, but he has only k signs. Help the minister to solve his problem, otherwise the poor guy can lose not only his position, but also his head.

More formally, every road in the Khanate is a line on the Oxy plane, given by an equation of the form $Ax + By + C = 0$ (A and B are not equal to 0 at the same time). You are required to determine whether you can put signs in at most k points so that each road had at least one sign installed.

Input

The input starts with two positive integers n, k ($1 \le n \le 10^5, 1 \le k \le 5$)

Next n lines contain three integers each, A_i, B_i, C_i, the coefficients of the equation that determines the road ($|A_i|, |B_i|, |C_i| \le 10^5$, $A_i^2 + B_i^2 \neq 0$).

It is guaranteed that no two roads coincide.

Output

If there is no solution, print "NO" in the single line (without the quotes).

Otherwise, print in the first line "YES" (without the quotes).

In the second line print a single number m ($m \le k$) — the number of used signs. In the next m lines print the descriptions of their locations.

Description of a location of one sign is two integers v, u. If u and v are two distinct integers between 1 and n, we assume that sign is at the point of intersection of roads number v and u. If $u = -1$, and v is an integer between 1 and n, then the sign is on the road number v in the point not lying on any other road. In any other case the description of a sign will be assumed invalid and your answer will be considered incorrect. In case if $v = u$, or if v and u are the numbers of two non-intersecting roads, your answer will also be considered incorrect.

The roads are numbered starting from 1 in the order in which they follow in the input.

Examples

input

```
3 1
1 0 0
0 -1 0
7 -93 0
```

output

```
YES
1
1 2
```

input

```
3 1
1 0 0
0 1 0
1 1 3
```

output

```
NO
```

input

```
2 3
3 4 5
5 6 7
```

output

```
YES
2
1 -1
2 -1
```

Note

Note that you do not have to minimize m, but it shouldn't be more than k.

In the first test all three roads intersect at point (0,0).

In the second test all three roads form a triangle and there is no way to place one sign so that it would stand on all three roads at once.

E. Longest Increasing Subsequence

time limit per test: 1.5 seconds
memory limit per test: 128 megabytes
input: standard input
output: standard output

Note that the memory limit in this problem is less than usual.

Let's consider an array consisting of positive integers, some positions of which contain gaps.

We have a collection of numbers that can be used to fill the gaps. Each number from the given collection can be used at most once.

Your task is to determine such way of filling gaps that the longest increasing subsequence in the formed array has a maximum size.

Input

The first line contains a single integer n — the length of the array ($1 \le n \le 10^5$).

The second line contains n space-separated integers — the elements of the sequence. A gap is marked as "-1". The elements that are not gaps are positive integers not exceeding 10^9. It is guaranteed that the sequence contains $0 \le k \le 1000$ gaps.

The third line contains a single positive integer m — the number of elements to fill the gaps ($k \le m \le 10^5$).

The fourth line contains m positive integers — the numbers to fill gaps. Each number is a positive integer not exceeding 10^9. Some numbers may be equal.

Output

Print n space-separated numbers in a single line — the resulting sequence. If there are multiple possible answers, print any of them.

Examples

input

```
3
1 2 3
1
10
```

output

```
1 2 3
```

input

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

output

```
1 2 3
```

input

```
2
-1 2
2
2 4
```

output

```
2 2
```

input

```
3
-1 -1 -1
5
1 1 1 1 2
```

output

```
1 1 2
```

input

```
4
-1 -1 -1 2
4
1 1 2 2
```

output

```
1 2 1 2
```

Note

In the first sample there are no gaps, so the correct answer is the initial sequence.

In the second sample there is only one way to get an increasing subsequence of length 3.

In the third sample answer "4 2" would also be correct. Note that only strictly increasing subsequences are considered.

In the fifth sample the answer "1 1 1 2" is not considered correct, as number 1 can be used in replacing only two times.