CODEFORCES β
Sponsored by Telegram

Mashup ITA Training #1

A. Robot Vacuum Cleaner

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

Pushok the dog has been chasing Imp for a few hours already.

Fortunately, Imp knows that Pushok is afraid of a robot vacuum cleaner.

While moving, the robot generates a string t consisting of letters 's' and 'h', that produces a lot of noise. We define *noise* of string t as the number of occurrences of string "sh" as a **subsequence** in it, in other words, the number of such pairs (i, j), that $i < j$ and $t_i = \mathrm{s}$ and $t_j = \mathrm{h}$.

The robot is off at the moment. Imp knows that it has a sequence of strings t_i in its memory, and he can arbitrary change their order. When the robot is started, it generates the string t as a concatenation of these strings in the given order. The noise of the resulting string equals the noise of this concatenation.

Help Imp to find the maximum noise he can achieve by changing the order of the strings.

Input

The first line contains a single integer n ($1 \le n \le 10^5$) — the number of strings in robot's memory.

Next n lines contain the strings $t_1, t_2, \ldots, t_n$, one per line. It is guaranteed that the strings are non-empty, contain only English letters 's' and 'h' and their total length does not exceed 10^5.

Output

Print a single integer — the maxumum possible *noise* Imp can achieve by changing the order of the strings.

Examples

input	Copy
4 ssh hs s hhhs	

output
18

input	Copy
2 h s	

output
1

Note

The optimal concatenation in the first sample is $ssshhshhhs$.

B. Recursive Queries

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

Let us define two functions f and g on positive integer numbers.

$$f(n) = \text{product of non-zero digits of } n$$
$$g(n) = \begin{cases} n & \text{if } n < 10 \\ g(f(n)) & \text{otherwise} \end{cases}$$

You need to process Q queries. In each query, you will be given three integers l, r and k. You need to print the number of integers x between l and r inclusive, such that $g(x) = k$.

Input

The first line of the input contains an integer Q ($1 \le Q \le 2 \times 10^5$) representing the number of queries.

Q lines follow, each of which contains 3 integers l, r and k ($1 \le l \le r \le 10^6$, $1 \le k \le 9$).

Output

For each query, print a single line containing the answer for that query.

Examples

input Copy

```
4
22 73 9
45 64 6
47 55 7
2 62 4
```

output

```
1
4
0
8
```

input Copy

```
4
82 94 6
56 67 4
28 59 9
39 74 4
```

output

```
3
1
1
5
```

Note

In the first example:

- $g(33) = 9$ as $g(33) = g(3 \times 3) = g(9) = 9$
- $g(47) = g(48) = g(60) = g(61) = 6$
- There are no such integers between 47 and 55.
- $g(4) = g(14) = g(22) = g(27) = g(39) = g(40) = g(41) = g(58) = 4$

C. A Lot of Games

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

Andrew, Fedor and Alex are inventive guys. Now they invent the game with strings for two players.

Given a group of n non-empty strings. During the game two players build the word together, initially the word is empty. The players move in turns. On his step player must add a single letter in the end of the word, the resulting word must be prefix of at least one string from the group. A player loses if he cannot move.

Andrew and Alex decided to play this game k times. The player who is the loser of the i-th game makes the first move in the $(i + 1)$-th game. Guys decided that the winner of all games is the player who wins the last (k-th) game. Andrew and Alex already started the game. Fedor wants to know who wins the game if both players will play optimally. Help him.

Input

The first line contains two integers, n and k ($1 \le n \le 10^5$; $1 \le k \le 10^9$).

Each of the next n lines contains a single non-empty string from the given group. The total length of all strings from the group doesn't exceed 10^5. Each string of the group consists only of lowercase English letters.

Output

If the player who moves first wins, print "`First`", otherwise print "`Second`" (without the quotes).

Examples

input	Copy
2 3 a b	

output
First

input	Copy
3 1 a b c	

output
First

input	Copy
1 2 ab	

output
Second

D. Chloe and the sequence

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

Chloe, the same as Vladik, is a competitive programmer. She didn't have any problems to get to the olympiad like Vladik, but she was confused by the task proposed on the olympiad.

Let's consider the following algorithm of generating a sequence of integers. Initially we have a sequence consisting of a single element equal to 1. Then we perform $(n - 1)$ steps. On each step we take the sequence we've got on the previous step, append it to the end of itself and insert in the middle the minimum positive integer we haven't used before. For example, we get the sequence $[1, 2, 1]$ after the first step, the sequence $[1, 2, 1, 3, 1, 2, 1]$ after the second step.

The task is to find the value of the element with index k (the elements are numbered from 1) in the obtained sequence, i. e. after $(n - 1)$ steps.

Please help Chloe to solve the problem!

Input

The only line contains two integers n and k ($1 \le n \le 50$, $1 \le k \le 2^n - 1$).

Output

Print single integer — the integer at the k-th position in the obtained sequence.

Examples

input	Copy
3 2	

output
2

input	Copy
4 8	

output
4

Note

In the first sample the obtained sequence is $[1, 2, 1, 3, 1, 2, 1]$. The number on the second position is 2.

In the second sample the obtained sequence is $[1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1]$. The number on the eighth position is 4.

E. Magic Numbers

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

Consider the decimal presentation of an integer. Let's call a number `d-magic` if digit d appears in decimal presentation of the number on even positions and nowhere else.

For example, the numbers $1727374, 17, 1$ are `7-magic` but $77, 7, 123, 34, 71$ are not `7-magic`. On the other hand the number 7 is `0-magic`, 123 is `2-magic`, 34 is `4-magic` and 71 is `1-magic`.

Find the number of `d-magic` numbers in the segment $[a, b]$ that are multiple of m. Because the answer can be very huge you should only find its value modulo $10^9 + 7$ (so you should find the remainder after dividing by $10^9 + 7$).

Input

The first line contains two integers m, d ($1 \le m \le 2000, 0 \le d \le 9$) — the parameters from the problem statement.

The second line contains positive integer a in decimal presentation (without leading zeroes).

The third line contains positive integer b in decimal presentation (without leading zeroes).

It is guaranteed that $a \le b$, the number of digits in a and b are the same and don't exceed 2000.

Output

Print the only integer a — the remainder after dividing by $10^9 + 7$ of the number of `d-magic` numbers in segment $[a, b]$ that are multiple of m.

Examples

input Copy

```
2 6
10
99
```

output

```
8
```

input Copy

```
2 0
1
9
```

output

```
4
```

input Copy

```
19 7
1000
9999
```

output

```
6
```

Note

The numbers from the answer of the first example are $16, 26, 36, 46, 56, 76, 86$ and 96.

The numbers from the answer of the second example are $2, 4, 6$ and 8.

The numbers from the answer of the third example are $1767, 2717, 5757, 6707, 8797$ and 9747.

F. Cloning Toys

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

Imp likes his plush toy a lot.

Recently, he found a machine that can clone plush toys. Imp knows that if he applies the machine to an original toy, he additionally gets one more original toy and one copy, and if he applies the machine to a copied toy, he gets two additional copies.

Initially, Imp has only one original toy. He wants to know if it is possible to use machine to get exactly x **copied** toys and y **original** toys? He can't throw toys away, and he can't apply the machine to a copy if he doesn't currently have any copies.

Input

The only line contains two integers x and y ($0 \le x, y \le 10^9$) — the number of copies and the number of original toys Imp wants to get (including the initial one).

Output

Print "Yes", if the desired configuration is possible, and "No" otherwise.

You can print each letter in arbitrary case (upper or lower).

Examples

input	Copy
6 3	

output
Yes

input	Copy
4 2	

output
No

input	Copy
1000 1001	

output
Yes

Note

In the first example, Imp has to apply the machine twice to original toys and then twice to copies.

G. Fools and Roads

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

They say that Berland has exactly two problems, fools and roads. Besides, Berland has n cities, populated by the fools and connected by the roads. All Berland roads are bidirectional. As there are many fools in Berland, between each pair of cities there is a path (or else the fools would get upset). Also, between each pair of cities there is no more than one simple path (or else the fools would get lost).

But that is not the end of Berland's special features. In this country fools sometimes visit each other and thus spoil the roads. The fools aren't very smart, so they always use only the simple paths.

A simple path is the path which goes through every Berland city not more than once.

The Berland government knows the paths which the fools use. Help the government count for each road, how many distinct fools can go on it.

Note how the fools' paths are given in the input.

Input

The first line contains a single integer n $(2 \le n \le 10^5)$ — the number of cities.

Each of the next $n - 1$ lines contains two space-separated integers u_i, v_i $(1 \le u_i, v_i \le n, u_i \ne v_i)$, that means that there is a road connecting cities u_i and v_i.

The next line contains integer k $(0 \le k \le 10^5)$ — the number of pairs of fools who visit each other.

Next k lines contain two space-separated numbers. The i-th line $(i > 0)$ contains numbers a_i, b_i $(1 \le a_i, b_i \le n)$. That means that the fool number $2i - 1$ lives in city a_i and visits the fool number $2i$, who lives in city b_i. The given pairs describe simple paths, because between every pair of cities there is only one simple path.

Output

Print $n - 1$ integer. The integers should be separated by spaces. The i-th number should equal the number of fools who can go on the i-th road. The roads are numbered starting from one in the order, in which they occur in the input.

Examples

input Copy

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

output

```
2 1 1 1
```

input Copy

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

output

```
3 1 1 1
```

Note

In the first sample the fool number one goes on the first and third road and the fool number 3 goes on the second, first and fourth ones.

In the second sample, the fools number 1, 3 and 5 go on the first road, the fool number 5 will go on the second road, on the third road goes the fool number 3, and on the fourth one goes fool number 1.