## Codeforces Round #427 (Div. 2)

# A. Key races

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

Two boys decided to compete in text typing on the site "Key races". During the competition, they have to type a text consisting of $s$ characters. The first participant types one character in $v_1$ milliseconds and has ping $t_1$ milliseconds. The second participant types one character in $v_2$ milliseconds and has ping $t_2$ milliseconds.

If connection ping (delay) is $t$ milliseconds, the competition passes for a participant as follows:

1. Exactly after $t$ milliseconds after the start of the competition the participant receives the text to be entered.
2. Right after that he starts to type it.
3. Exactly $t$ milliseconds after he ends typing all the text, the site receives information about it.

The winner is the participant whose information on the success comes earlier. If the information comes from both participants at the same time, it is considered that there is a draw.

Given the length of the text and the information about participants, determine the result of the game.

### Input

The first line contains five integers $s$, $v_1$, $v_2$, $t_1$, $t_2$ ($1 \le s, v_1, v_2, t_1, t_2 \le 1000$) — the number of characters in the text, the time of typing one character for the first participant, the time of typing one character for the the second participant, the ping of the first participant and the ping of the second participant.

### Output

If the first participant wins, print "`First`". If the second participant wins, print "`Second`". In case of a draw print "`Friendship`".

**Examples**

| input |
| --- |
| 5 1 2 1 2 |
| **output** |
| First |

| input |
| --- |
| 3 3 1 1 1 |
| **output** |
| Second |

| input |
| --- |
| 4 5 3 1 5 |
| **output** |
| Friendship |

### Note

In the first example, information on the success of the first participant comes in $7$ milliseconds, of the second participant — in $14$ milliseconds. So, the first wins.

In the second example, information on the success of the first participant comes in $11$ milliseconds, of the second participant — in $5$ milliseconds. So, the second wins.

In the third example, information on the success of the first participant comes in $22$ milliseconds, of the second participant — in $22$ milliseconds. So, it is be a draw.

# B. The number on the board

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

Some natural number was written on the board. Its sum of digits was not less than $k$. But you were distracted a bit, and someone changed this number to $n$, replacing some digits with others. It's known that the length of the number didn't change.

You have to find the minimum number of digits in which these two numbers can differ.

## Input

The first line contains integer $k$ ($1 \le k \le 10^9$).

The second line contains integer $n$ ($1 \le n < 10^{100000}$).

There are no leading zeros in $n$. It's guaranteed that this situation is possible.

## Output

Print the minimum number of digits in which the initial number and $n$ can differ.

**Examples**

| input |
| --- |
| 3<br>11 |
| **output** |
| 1 |

| input |
| --- |
| 3<br>99 |
| **output** |
| 0 |

## Note

In the first example, the initial number could be $12$.

In the second example the sum of the digits of $n$ is not less than $k$. The initial number could be equal to $n$.

# C. Star sky

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

The Cartesian coordinate system is set in the sky. There you can see $n$ stars, the $i$-th has coordinates $(x_i, y_i)$, a maximum brightness $c$, equal for all stars, and an initial brightness $s_i$ $(0 \le s_i \le c)$.

Over time the stars twinkle. At moment $0$ the $i$-th star has brightness $s_i$. Let at moment $t$ some star has brightness $x$. Then at moment $(t+1)$ this star will have brightness $x+1$, if $x+1 \le c$, and $0$, otherwise.

You want to look at the sky $q$ times. In the $i$-th time you will look at the moment $t_i$ and you will see a rectangle with sides parallel to the coordinate axes, the lower left corner has coordinates $(x_{1i}, y_{1i})$ and the upper right — $(x_{2i}, y_{2i})$. For each view, you want to know the total brightness of the stars lying in the viewed rectangle.

A star lies in a rectangle if it lies on its border or lies strictly inside it.

## Input

The first line contains three integers $n, q, c$ $(1 \le n, q \le 10^5, 1 \le c \le 10)$ — the number of the stars, the number of the views and the maximum brightness of the stars.

The next $n$ lines contain the stars description. The $i$-th from these lines contains three integers $x_i, y_i, s_i$ $(1 \le x_i, y_i \le 100, 0 \le s_i \le c \le 10)$ — the coordinates of $i$-th star and its initial brightness.

The next $q$ lines contain the views description. The $i$-th from these lines contains five integers $t_i, x_{1i}, y_{1i}, x_{2i}, y_{2i}$ $(0 \le t_i \le 10^9, 1 \le x_{1i} < x_{2i} \le 100, 1 \le y_{1i} < y_{2i} \le 100)$ — the moment of the $i$-th view and the coordinates of the viewed rectangle.

## Output

For each view print the total brightness of the viewed stars.

**Examples**

| input |
| --- |
| 2 3 3<br>1 1 1<br>3 2 0<br>2 1 1 2 2<br>0 2 1 4 5<br>5 1 1 5 5 |

| output |
| --- |
| 3<br>0<br>3 |

| input |
| --- |
| 3 4 5<br>1 1 2<br>2 3 0<br>3 3 1<br>0 1 1 100 100<br>1 2 2 4 4<br>2 2 1 4 7<br>1 50 50 51 51 |

| output |
| --- |
| 3<br>3<br>5<br>0 |

## Note

Let's consider the first example.

At the first view, you can see only the first star. At moment $2$ its brightness is $3$, so the answer is $3$.

At the second view, you can see only the second star. At moment $0$ its brightness is $0$, so the answer is $0$.

At the third view, you can see both stars. At moment $5$ brightness of the first is $2$, and brightness of the second is $1$, so the answer is $3$.

# D. Palindromic characteristics

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

Palindromic characteristics of string $s$ with length $|s|$ is a sequence of $|s|$ integers, where $k$-th number is the total number of non-empty substrings of $s$ which are $k$-palindromes.

A string is $1$-palindrome if and only if it reads the same backward as forward.

A string is $k$-palindrome ($k > 1$) if and only if:

1. Its left half equals to its right half.
2. Its left and right halfs are non-empty ($k - 1$)-palindromes.

The left half of string $t$ is its prefix of length $\lfloor |t| / 2 \rfloor$, and right half — the suffix of the same length. $\lfloor |t| / 2 \rfloor$ denotes the length of string $t$ divided by $2$, rounded down.

Note that each substring is counted as many times as it appears in the string. For example, in the string "aaa" the substring "a" appears 3 times.

## Input

The first line contains the string $s$ ($1 \le |s| \le 5000$) consisting of lowercase English letters.

## Output

Print $|s|$ integers — palindromic characteristics of string $s$.

**Examples**

| input |
| --- |
| abba |
| **output** |
| 6 1 0 0 |

| input |
| --- |
| abacaba |
| **output** |
| 12 4 1 0 0 0 0 |

## Note

In the first example 1-palindromes are substring «a», «b», «b», «a», «bb», «abba», the substring «bb» is 2-palindrome. There are no 3- and 4-palindromes here.

# E. The penguin's game

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

*Pay attention: this problem is interactive.*

Penguin Xoriy came up with a new game recently. He has $n$ icicles numbered from $1$ to $n$. Each icicle has a temperature — an integer from $1$ to $10^9$. **Exactly two** of these icicles are special: their temperature is $y$, while a temperature of all the others is $x \neq y$. You have to find those special icicles. You can choose a *non-empty* subset of icicles and ask the penguin what is the bitwise exclusive OR (`XOR`) of the temperatures of the icicles in this subset. Note that you can't ask more than **19** questions.

You are to find the special icicles.

## Input

The first line contains three integers $n, x, y$ ($2 \leq n \leq 1000$, $1 \leq x, y \leq 10^9$, $x \neq y$) — the number of icicles, the temperature of non-special icicles and the temperature of the special icicles.

## Output

To give your answer to the penguin you have to print character "!" (without quotes), then print two integers $p_1, p_2$ ($p_1 < p_2$) — the indexes of the special icicles **in ascending order**. Note that "!" and $p_1$ should be separated by a space; the indexes should be separated by a space too. After you gave the answer your program should terminate immediately.

## Interaction

To ask a question print character "?" (without quotes), an integer $c$ ($1 \leq c \leq n$), and $c$ distinct integers $p_1, p_2, ..., p_c$ ($1 \leq p_i \leq n$) — the indexes of icicles that you want to know about. Note that "?" and $c$ should be separated by a space; the indexes should be separated by a space too.

After you asked the question, read a single integer — the answer.

Note that you can't ask more than **19** questions. If you ask more than 19 questions or at least one incorrect question, your solution will get "`Wrong answer`".

If at some moment your program reads $-1$ as an answer, it should immediately exit (for example, by calling exit(0)). You will get "`Wrong answer`" in this case, it means that you asked more than 19 questions, or asked an invalid question. If you ignore this, you can get other verdicts since your program will continue to read from a closed stream.

Your solution will get "`Idleness Limit Exceeded`", if you don't print anything or forget to flush the output, including for the final answer .

To flush you can use (just after printing):

- `fflush(stdout)` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- `flush(output)` in Pascal;
- For other languages see the documentation.

### Hacking

For hacking use the following format:

$n\ x\ y\ p_1\ p_2$

Here $1 \leq p_1 < p_2 \leq n$ are the indexes of the special icicles.

Contestant programs will not be able to see this input.

### Example

| input |
| --- |
| 4 2 1<br>2<br>1<br>1 |
| **output** |
| ? 3 1 2 3<br>? 1 1<br>? 1 3<br>! 1 3 |

## Note

The answer for the first question is $1 \oplus 2 \oplus 1 = 2$.

The answer for the second and the third questions is 1, therefore, special icicles are indexes 1 and 3.

You can read more about bitwise `XOR` operation here: https://en.wikipedia.org/wiki/Bitwise_operation#XOR.

# F. Roads in the Kingdom

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

In the Kingdom K., there are $n$ towns numbered with integers from $1$ to $n$. The towns are connected by $n$ bi-directional roads numbered with integers from $1$ to $n$. The $i$-th road connects the towns $u_i$ and $v_i$ and its length is $l_i$. There is no more than one road between two towns. Also, there are no roads that connect the towns with itself.

Let's call the inconvenience of the roads the maximum of the shortest distances between all pairs of towns.

Because of lack of money, it was decided to close down one of the roads so that after its removal it is still possible to reach any town from any other. You have to find the minimum possible inconvenience of the roads after closing down one of the roads.

## Input

The first line contains the integer $n$ $(3 \le n \le 2 \cdot 10^5)$ — the number of towns and roads.

The next $n$ lines contain the roads description. The $i$-th from these lines contains three integers $u_i$, $v_i$, $l_i$ $(1 \le u_i, v_i \le n, 1 \le l_i \le 10^9)$ — the numbers of towns connected by the $i$-th road and the length of the $i$-th road. No road connects a town to itself, no two roads connect the same towns.

*It's guaranteed that it's always possible to close down one of the roads so that all the towns are still reachable from each other.*

## Output

Print a single integer — the minimum possible inconvenience of the roads after the refusal from one of the roads.

**Examples**

| input |
| --- |
| 3<br>1 2 4<br>2 3 5<br>1 3 1 |

| output |
| --- |
| 5 |

| input |
| --- |
| 5<br>2 3 7<br>3 1 9<br>4 1 8<br>3 5 4<br>4 5 5 |

| output |
| --- |
| 18 |