Educational Codeforces Round 17

A. k-th divisor

2 seconds, 256 megabytes

You are given two integers n and k. Find k-th smallest divisor of n, or report that it doesn't exist.

Divisor of n is any such natural number, that n can be divided by it without remainder.

Input

The first line contains two integers n and k ($1 \le n \le 10^{15}$, $1 \le k \le 10^9$).

Output

If n has less than k divisors, output `-1`.

Otherwise, output the k-th smallest divisor of n.

input
4 2
output
2

input
5 3
output
-1

input
12 5
output
6

In the first example, number 4 has three divisors: 1, 2 and 4. The second one is 2.

In the second example, number 5 has only two divisors: 1 and 5. The third divisor doesn't exist, so the answer is `-1`.

B. USB vs. PS/2

2 seconds, 256 megabytes

Due to the increase in the number of students of Berland State University it was decided to equip a new computer room. You were given the task of buying mouses, and you have to spend as little as possible. After all, the country is in crisis!

The computers bought for the room were different. Some of them had only USB ports, some — only PS/2 ports, and some had both options.

You have found a price list of a certain computer shop. In it, for m mouses it is specified the cost and the type of the port that is required to plug the mouse in (USB or PS/2). Each mouse from the list can be bought at most once.

You want to buy some set of mouses from the given price list in such a way so that you maximize the number of computers equipped with mouses (it is not guaranteed that you will be able to equip all of the computers), and in case of equality of this value you want to minimize the total cost of mouses you will buy.

Input

The first line contains three integers a, b and c ($0 \le a, b, c \le 10^5$) — the number of computers that only have USB ports, the number of computers, that only have PS/2 ports, and the number of computers, that have both options, respectively.

The next line contains one integer m ($0 \le m \le 3 \cdot 10^5$) — the number of mouses in the price list.

The next m lines each describe another mouse. The i-th line contains first integer val_i ($1 \le val_i \le 10^9$) — the cost of the i-th mouse, then the type of port (USB or PS/2) that is required to plug the mouse in.

Output

Output two integers separated by space — the number of equipped computers and the total cost of the mouses you will buy.

input
```
2 1 1
4
5 USB
6 PS/2
3 PS/2
7 PS/2
```
output
```
3 14
```

In the first example you can buy the first three mouses. This way you will equip one of the computers that has only a USB port with a USB mouse, and the two PS/2 mouses you will plug into the computer with PS/2 port and the computer with both ports.

C. Two strings

2 seconds, 256 megabytes

You are given two strings a and b. You have to remove the minimum possible number of **consecutive** (standing one after another) characters from string b in such a way that it becomes a subsequence of string a. It can happen that you will not need to remove any characters at all, or maybe you will have to remove all of the characters from b and make it empty.

Subsequence of string s is any such string that can be obtained by erasing zero or more characters (**not necessarily consecutive**) from string s.

Input
The first line contains string a, and the second line — string b. Both of these strings are nonempty and consist of lowercase letters of English alphabet. The length of each string is no bigger than 10^5 characters.

Output
On the first line output a subsequence of string a, obtained from b by erasing the minimum number of consecutive characters.

If the answer consists of zero characters, output «-» (a minus sign).

input
```
hi
bob
```
output
```
-
```

input
```
abca
accepted
```
output
```
ac
```

input
```
abacaba
abcdcba
```
output
```
abcba
```

In the first example strings a and b don't share any symbols, so the longest string that you can get is empty.

In the second example `ac` is a subsequence of a, and at the same time you can obtain it by erasing consecutive symbols `cepted` from string b.

D. Maximum path

1 second, 256 megabytes

You are given a rectangular table $3 \times n$. Each cell contains an integer. You can move from one cell to another if they share a side.

Find such path from the upper left cell to the bottom right cell of the table that doesn't visit any of the cells twice, and the sum of numbers written in the cells of this path is maximum possible.

Input
The first line contains an integer n $(1 \le n \le 10^5)$ — the number of columns in the table.

Next three lines contain n integers each — the description of the table. The j-th number in the i-th line corresponds to the cell a_{ij} $(-10^9 \le a_{ij} \le 10^9)$ of the table.

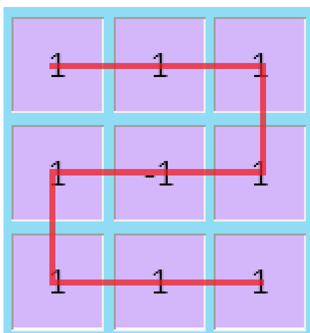
Output

Output the maximum sum of numbers on a path from the upper left cell to the bottom right cell of the table, that doesn't visit any of the cells twice.

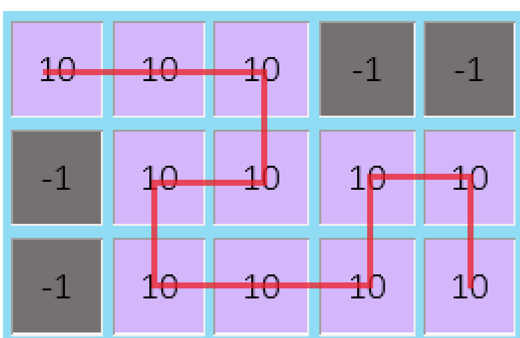
input
```
3
1 1 1
1 -1 1
1 1 1
```
output
```
7
```

input
```
5
10 10 10 -1 -1
-1 10 10 10 10
-1 10 10 10 10
```
output
```
110
```

The path for the first example:



The path for the second example:



E. Radio stations

2 seconds, 256 megabytes

In the lattice points of the coordinate line there are n radio stations, the i-th of which is described by three integers:

- x_i — the coordinate of the i-th station on the line,
- r_i — the broadcasting range of the i-th station,
- f_i — the broadcasting frequency of the i-th station.

We will say that two radio stations with numbers i and j reach each other, if the broadcasting range of each of them is more or equal to the distance between them. In other words $min(r_i, r_j) \geq |x_i - x_j|$.

Let's call a pair of radio stations (i, j) bad if $i < j$, stations i and j reach each other and they are close in frequency, that is, $|f_i - f_j| \leq k$.

Find the number of bad pairs of radio stations.

Input

The first line contains two integers n and k $(1 \leq n \leq 10^5, 0 \leq k \leq 10)$ — the number of radio stations and the maximum difference in the frequencies for the pair of stations that reach each other to be considered bad.

In the next n lines follow the descriptions of radio stations. Each line contains three integers x_i, r_i and f_i $(1 \leq x_i, r_i \leq 10^9, 1 \leq f_i \leq 10^4)$ — the coordinate of the i-th radio station, it's broadcasting range and it's broadcasting frequency. **No two radio stations will share a coordinate**.

Output

Output the number of bad pairs of radio stations.

input
```
3 2
1 3 10
3 2 5
4 10 8
```
output
```
1
```

input

```
3 3
1 3 10
3 2 5
4 10 8
```

output

```
2
```

input

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

output

```
2
```

input

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

output

```
5
```

F. Tree nesting

2 seconds, 256 megabytes

You are given two trees (connected undirected acyclic graphs) S and T.

Count the number of subtrees (connected subgraphs) of S that are isomorphic to tree T. Since this number can get quite large, output it modulo $10^9 + 7$.

Two subtrees of tree S are considered different, if there exists a vertex in S that belongs to exactly one of them.

Tree G is called isomorphic to tree H if there exists a bijection f from the set of vertices of G to the set of vertices of H that has the following property: if there is an edge between vertices A and B in tree G, then there must be an edge between vertices $f(A)$ and $f(B)$ in tree H. And vice versa — if there is an edge between vertices A and B in tree H, there must be an edge between $f^{-1}(A)$ and $f^{-1}(B)$ in tree G.

Input

The first line contains a single integer $|S|$ ($1 \le |S| \le 1000$) — the number of vertices of tree S.

Next $|S| - 1$ lines contain two integers u_i and v_i ($1 \le u_i, v_i \le |S|$) and describe edges of tree S.

The next line contains a single integer $|T|$ ($1 \le |T| \le 12$) — the number of vertices of tree T.

Next $|T| - 1$ lines contain two integers x_i and y_i ($1 \le x_i, y_i \le |T|$) and describe edges of tree T.

Output

On the first line output a single integer — the answer to the given task modulo $10^9 + 7$.

input

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

output

```
3
```

input

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

output

```
1
```

input

```
7
1 2
1 3
1 4
1 5
1 6
1 7
4
4 1
4 2
4 3
```

output

```
20
```

input

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

output

```
0
```