Codeforces Sponsored by Telegram

Codeforces Round #366 (Div. 1)

A. Thor

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

Thor is getting used to the Earth. As a gift Loki gave him a smartphone. There are n applications on this phone. Thor is fascinated by this phone. He has only one minor issue: he can't count the number of unread notifications generated by those applications (maybe Loki put a curse on it so he can't).

q events are about to happen (in chronological order). They are of three types:

1. Application x generates a notification (this new notification is unread).
2. Thor reads all notifications generated so far by application x (he may re-read some notifications).
3. Thor reads the first t notifications generated by phone applications (notifications generated in first t events of the first type). It's guaranteed that there were at least t events of the first type before this event. Please note that he doesn't read first t unread notifications, he just reads the very first t notifications generated on his phone and he may re-read some of them in this operation.

Please help Thor and tell him the number of unread notifications after each event. You may assume that initially there are no notifications in the phone.

Input

The first line of input contains two integers n and q $(1 \le n, q \le 300\,000)$ — the number of applications and the number of events to happen.

The next q lines contain the events. The i-th of these lines starts with an integer $type_i$ — type of the i-th event. If $type_i = 1$ or $type_i = 2$ then it is followed by an integer x_i. Otherwise it is followed by an integer t_i $(1 \le type_i \le 3, 1 \le x_i \le n, 1 \le t_i \le q)$.

Output

Print the number of unread notifications after each event.

Examples

input

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

output

```
1
2
3
2
```

input

```
4 6
1 2
1 4
1 2
3 3
1 3
1 3
```

output

```
1
2
3
0
1
2
```

Note

In the first sample:

1. Application 3 generates a notification (there is 1 unread notification).
2. Application 1 generates a notification (there are 2 unread notifications).
3. Application 2 generates a notification (there are 3 unread notifications).
4. Thor reads the notification generated by application 3, there are 2 unread notifications left.

In the second sample test:

1. Application 2 generates a notification (there is 1 unread notification).
2. Application 4 generates a notification (there are 2 unread notifications).
3. Application 2 generates a notification (there are 3 unread notifications).
4. Thor reads first three notifications and since there are only three of them so far, there will be no unread notification left.
5. Application 3 generates a notification (there is 1 unread notification).
6. Application 3 generates a notification (there are 2 unread notifications).

B. Ant Man

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Scott Lang is at war with Darren Cross. There are n chairs in a hall where they are, numbered with $1, 2, ..., n$ from left to right. The i-th chair is located at coordinate x_i. Scott is on chair number s and Cross is on chair number e. Scott can jump to all other chairs (not only neighboring chairs). He wants to start at his position (chair number s), visit each chair **exactly once** and end up on chair number e with Cross.

As we all know, Scott can shrink or grow big (grow big only to his normal size), so at any moment of time he can be either small or large (normal). The thing is, he can only shrink or grow big while being on a chair (not in the air while jumping to another chair). Jumping takes time, but shrinking and growing big takes no time. Jumping from chair number i to chair number j takes $|x_i - x_j|$ seconds. Also, jumping off a chair and landing on a chair takes extra amount of time.

If Scott wants to jump to a chair on his left, he can only be small, and if he wants to jump to a chair on his right he should be large.

Jumping off the i-th chair takes:

- c_i extra seconds if he's small.
- d_i extra seconds otherwise (he's large).

Also, landing on i-th chair takes:

- b_i extra seconds if he's small.
- a_i extra seconds otherwise (he's large).

In simpler words, jumping from i-th chair to j-th chair takes exactly:

- $|x_i - x_j| + c_i + b_j$ seconds if $j < i$.
- $|x_i - x_j| + d_i + a_j$ seconds otherwise ($j > i$).

Given values of x, a, b, c, d find the minimum time Scott can get to Cross, assuming he wants to visit each chair exactly once.

Input

The first line of the input contains three integers n, s and e ($2 \le n \le 5000$, $1 \le s, e \le n$, $s \ne e$) — the total number of chairs, starting and ending positions of Scott.

The second line contains n integers $x_1, x_2, ..., x_n$ ($1 \le x_1 < x_2 < ... < x_n \le 10^9$).

The third line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_1, a_2, ..., a_n \le 10^9$).

The fourth line contains n integers $b_1, b_2, ..., b_n$ ($1 \le b_1, b_2, ..., b_n \le 10^9$).

The fifth line contains n integers $c_1, c_2, ..., c_n$ ($1 \le c_1, c_2, ..., c_n \le 10^9$).

The sixth line contains n integers $d_1, d_2, ..., d_n$ ($1 \le d_1, d_2, ..., d_n \le 10^9$).

Output

Print the minimum amount of time Scott needs to get to the Cross while visiting each chair exactly once.

Example

input

```
7 4 3
8 11 12 16 17 18 20
17 16 20 2 20 5 13
17 8 8 16 12 15 13
12 4 16 4 15 7 6
8 14 2 11 17 12 8
```

output

```
139
```

Note

In the sample testcase, an optimal solution would be . Spent time would be $17 + 24 + 23 + 20 + 33 + 22 = 139$.

C. Black Widow

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

Natalia Romanova is trying to test something on the new gun S.H.I.E.L.D gave her. In order to determine the result of the test, she needs to find the number of answers to a certain equation. The equation is of form:

Where represents logical OR and represents logical exclusive OR (XOR), and $v_{i,j}$ are some boolean variables or their negations. Natalia calls the left side of the equation a XNF formula. Each statement in brackets is called a clause, and $v_{i,j}$ are called literals.

In the equation Natalia has, the left side is actually a 2-XNF-2 containing variables $x_1, x_2, ..., x_m$ and their negations. An XNF formula is 2-XNF-2 if:

1. For each $1 \le i \le n$, $k_i \le 2$, i.e. the size of each clause doesn't exceed two.
2. Each variable occurs **in the formula at most two times** (with negation and without negation in total). Please note that it's possible that a variable occurs twice but its negation doesn't occur in any clause (or vice versa).

Natalia is given a formula of m variables, consisting of n clauses. Please, make sure to check the samples in order to properly understand how the formula looks like.

Natalia is more into fight than theory, so she asked you to tell her the number of answers to this equation. More precisely, you need to find the number of ways to set $x_1, ..., x_m$ with $true$ and $false$ (out of total of 2^m ways) so that the equation is satisfied. Since this number can be extremely large, you need to print the answer modulo $10^9 + 7$.

Please, note that some variable may appear twice in one clause, or not appear in the equation at all (but still, setting it to $false$ or $true$ gives different ways to set variables).

Input

The first line of input contains two integers n and m $(1 \le n, m \le 100\ 000)$ — the number of clauses and the number of variables respectively.

The next n lines contain the formula. The i-th of them starts with an integer k_i — the number of literals in the i-th clause. It is followed by k_i non-zero integers $a_{i,1}, ..., a_{i,k_i}$. If $a_{i,j} > 0$ then $v_{i,j}$ is $x_{a_{i,j}}$ otherwise it's negation of $x_{-a_{i,j}}$ $(1 \le k_i \le 2, -m \le a_{i,j} \le m, a_{i,j} \ne 0)$.

Output

Print the answer modulo $1\ 000\ 000\ 007$ $(10^9 + 7)$ in one line.

Examples

input

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

output

```
48
```

input

```
8 10
1 -5
2 4 -6
2 -2 -6
2 -7 9
2 10 -1
2 3 -1
2 -8 9
2 5 8
```

output

```
544
```

input

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

output

```
4
```

Note

The equation in the first sample is:

The equation in the second sample is:

The equation in the third sample is:

D. Captain America

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

Steve Rogers is fascinated with new vibranium shields S.H.I.E.L.D gave him. They're all uncolored. There are n shields in total, the i-th shield is located at point (x_i, y_i) of the coordinate plane. It's possible that two or more shields share the same location.

Steve wants to paint all these shields. He paints each shield in either red or blue. Painting a shield in red costs r dollars while painting it in blue costs b dollars.

Additionally, there are m constraints Steve wants to be satisfied. The i-th constraint is provided by three integers t_i, l_i and d_i:

- If $t_i = 1$, then the absolute difference between the number of red and blue shields on line $x = l_i$ should not exceed d_i.
- If $t_i = 2$, then the absolute difference between the number of red and blue shields on line $y = l_i$ should not exceed d_i.

Steve gave you the task of finding the painting that satisfies all the condition and the total cost is minimum.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 100\,000$) — the number of shields and the number of constraints respectively.

The second line contains two integers r and b ($1 \le r, b \le 10^9$).

The next n lines contain the shields coordinates. The i-th of these lines contains two integers x_i and y_i ($1 \le x_i, y_i \le 10^9$).

The next m lines contain the constrains. The j-th of these lines contains three integers t_j, l_j and d_j ($1 \le t_j \le 2$, $1 \le l_j \le 10^9$, $0 \le d_j \le n$).

Output

If satisfying all the constraints is impossible print `-1` in first and only line of the output.

Otherwise, print the minimum total cost in the first line of output. In the second line print a string of length n consisting of letters 'r' and 'b' only. The i-th character should be 'r' if the i-th shield should be painted red in the optimal answer and 'b' if it should be painted blue. The cost of painting shields in these colors should be equal the minimum cost you printed on the first line.

If there exist more than one optimal solution, print any of them.

Examples

input

```
5 6
8 3
2 10
1 5
9 10
9 10
2 8
1 9 1
1 2 1
2 10 3
2 10 2
1 1 1
2 5 2
```

output

```
25
rbrbb
```

input

```
4 4
7 3
10 3
9 8
10 3
2 8
2 8 0
2 8 0
1 2 0
1 9 0
```

output

```
-1
```

E. Iron Man

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Tony Stark is playing a game with his suits (they have auto-pilot now). He lives in Malibu. Malibu has n junctions numbered from 1 to n, connected with $n - 1$ roads. One can get from a junction to any other junction using these roads (graph of Malibu forms a tree).

Tony has m suits. There's a special plan for each suit. The i-th suit will appear at the moment of time t_i in the junction v_i, and will move to junction u_i using the shortest path between v_i and u_i with the speed c_i roads per second (passing a junctions takes no time), and vanishing immediately when arriving at u_i (if it reaches u_i in time q, it's available there at moment q, but not in further moments). Also, suits move continuously (for example if $v_i \neq u_i$, at time it's in the middle of a road. Please note that if $v_i = u_i$ it means the suit will be at junction number v_i only at moment t_i and then it vanishes.

An explosion happens if at any moment of time two suits share the same exact location (it may be in a junction or somewhere on a road; while appearing, vanishing or moving).

Your task is to tell Tony the moment of the the first explosion (if there will be any).

Input

The first line of the input contains two integers n and m ($1 \leq n, m \leq 100\,000$) — the number of junctions and the number of suits respectively.

The next $n - 1$ lines contain the roads descriptions. Each line contains two integers a_i and b_i — endpoints of the i-th road ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$).

The next m lines contain the suit descriptions. The i-th of them contains four integers t_i, c_i, v_i and u_i ($0 \leq t_i \leq 10\,000$, $1 \leq c_i \leq 10\,000$, $1 \leq v_i, u_i \leq n$), meaning the i-th suit will appear at moment of time t_i at the junction v_i and will move to the junction u_i with a speed c_i roads per second.

Output

If there would be no explosions at all, print `-1` in the first and only line of output.

Otherwise print the moment of the first explosion.

Your answer will be considered correct if its relative or absolute error doesn't exceed 10^{-6}.

Examples

input

```
6 4
2 5
6 5
3 6
4 6
4 1
27 6 1 3
9 5 1 6
27 4 3 4
11 29 2 6
```

output

```
27.3
```

input

```
6 4
3 1
4 5
6 4
6 1
2 6
16 4 4 5
13 20 6 2
3 16 4 5
28 5 3 5
```

output

```
-1
```