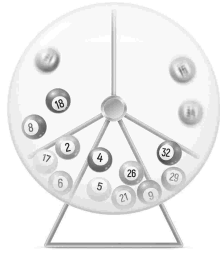
Croatian Open Competition in Informatics

Round 4, February 10th 2024

Tasks

Task	Time limit	Memory limit	Score
Bingo	1 second	512 MiB	50
Knjige	1 second	512 MiB	70
Lepeze	2 seconds	512 MiB	110
Putovanje	2 seconds	512 MiB	110
Roboti	2 seconds	512 MiB	110
Total			450

Task Bingo



It's time to play Bingo!

To play Bingo, you need a game master and a drum with 90 balls, each containing a number from 1 to 90, sucha that every number appears on exactly one ball.

Before the game starts, the game master gives each of the n players a board of size 5×5. Each field of the board contains an integer between 1 and 90, where all the integers on the board are distinct. Each player gets a unique board.

After the players receive their boards, the game can begin.

The game master starts drawing balls from the drum. After drawing a ball with the number x_i, he announces that number and puts the ball aside. The players then check their boards and, if they have the drawn number, they mark it.

When a player marks all the 5 numbers in a row, column, main diagonal or antidiagonal, he has a *Bingo!* and shouts it out. The game ends and that player wins.

To make the game more interesting, the game master has decided to introduce an additional rule. Namely, the game master will draw m balls from the drum before anyone is allowed to shout *Bingo!* (even if he has already marked all the numbers in a row, column, or diagonal).

But, as soon as the game master drew m balls, there was a commotion: all the players shouted *Bingo!* at the same time.

The game master is confused and doesn't know who to trust. To resolve this situation, he asked you for help. Determine which players had a *Bingo!* after drawing m balls, i.e. which players had all the numbers marked in at least one row, column, or diagonal.

Input

The first line contains the integer n ($1 \le n \le 50$), the number of players.

Then, n times six lines follows:

- The first of these lines contains a string of at most 20 lowercase English letters, the name of the player. No two players have the same name.
- Then five lines follow with five integers between 1 and 90, which describe the player's board. All the integers on the board are distinct.

The next line contains the integer m ($1 \le m \le 90$), the number of balls the game master drew before the players shouted *Bingo!*.

The next line contains a sequence of m integers between 1 and 90, the numbers the game master drew from the drum. Each number is drawn at most once.

Output

In the first line, output k, the number of players that had a *Bingo!* after drawing m balls.

In the next k lines, output the names of the players that had a *Bingo!* after drawing m balls. The names should be output in the same order as they appear in the input.

Scoring

Subtask	Points	Constraints
1	12	There is only one player, i. e. $n = 1$
2	22	At most one player will have *Bingo!*
3	16	No additional constraints.

Examples

input

```
3
babylasagna
10 11 12 13 14
15 16 17 18 19
20 21 22 23 24
25 26 27 28 29
30 31 32 33 34
nataliebalmix
10 20 30 40 50
11 21 31 41 51
12 22 32 42 52
13 23 33 43 53
14 24 34 44 54
lettri
89 88 87 86 10
85 84 83 11 82
81 80 12 79 78
77 13 76 75 74
14 73 72 71 70
6
10 11 12 13 14 15
```

output

```
3
babylasagna
nataliebalmix
lettri
```

input

```
1
honi
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
4
1 2 49 50
```

output

```
0
```

input

```
4
rim
15 23 14 26 34
12 11 13 16 17
90 67 45 24 18
85 82 77 66 22
62 71 32 35 7
tim
61 89 25 63 12
29 30 31 32 33
11 17 42 24 18
88 82 77 66 22
44 71 54 35 7
dagi
15 23 14 26 34
12 11 13 16 17
90 67 45 24 18
85 82 77 66 22
62 71 36 35 7
dim
15 23 14 26 34
12 11 13 16 17
90 67 45 24 18
85 82 77 66 22
42 51 32 33 7
7
15 11 66 7 42 30 61
```

output

```
1
tim
```

Clarification of the first example:
`babylasagna` has a *Bingo!* in the first row.
`nataliebalmix` has a *Bingo!* in the first column.
`lettri` has a *Bingo!* in the diagonal starting from the bottom-left corner to the top-right corner.

Clarification of the second example:
Only 4 balls were drawn, so no player can have marked all the 5 numbers in a row, column, or diagonal.

Task Knjige

Marko was at the *Interliber* book fair, and he bought n books. The attraction of the i-th book is k_i. Marko arranged the books on the shelf according to their attraction values, so the first book from the left is the least attractive, and every next one to the right is more or equally attractive than the previous one.

It has been quite some time since Interliber, but Marko has only now found time to read the books. He will spend a total of t minutes reading.

For each book, he can either read it in its entirety, which takes him a minutes; or read only the content from the covers, which takes him b minutes.

He will start from the leftmost book. After finishing the current book (either entirely or just the content from the covers), he moves on to the next book, which is the first one to the right of the book he just read. Marko's *inspiration* is equal to the sum of the attraction values of the books he has read in their entirety. What is the maximum value of Marko's *inspiration* after t minutes?

Note: If Marko starts reading a book but fails to finish it before the end of t minutes, that book does not contribute to his inspiration.

Input

The first line contains integers n, t, a and b ($1 \leq n \leq 2 \cdot 10^5, 1 \leq t \leq 10^9, 1 \leq b < a \leq 10^9$), the number of books, the time Marko will spend reading, time required for reading a book and the time required to read the content from the covers.

The second line contains n integers k_i ($1 \leq k_i \leq 10^9, k_i \leq k_{i+1}$), the attraction values of books.

Output

In the first and only line print Mirko's maximal *inspiration* after t minutes.

Scoring

Subtask	Points	Constraints
1	7	$k_i = k_{i+1}$ for each $i = 1, \ldots, n-1$
2	27	$n \leq 1\,000$
3	36	No additional constraints.

Examples

input

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

output

```
6
```

input

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

output

```
6
```

input

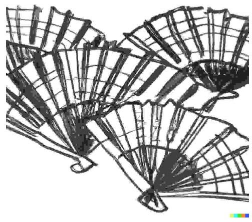
```
4 10 3 2
3 4 5 6
```

output

```
12
```

Clarification of the first example:
For example, Marko can read the first book in its entirety, read only the content from the covers of the second book, and read the third book in its entirety, thus achieving the maximum possible inspiration.

Task Lepeze



Little Fran received a wooden frame in the shape of a regular polygon as a gift. As polygon has n vertices, he also received $\frac{n(n-3)}{2}$ wooden sticks that match each possible diagonal. Vertices of the polygon are labelled with integers from 1 to n in counterclockwise order. In the beginning, Fran arranged $n-3$ sticks inside the frame in such a way that every stick touches two non-neighboring vertive of the frame, and no two sticks cross each other. In other words, he made a triangulation. As that was not interesting enough for him, he decided to play with this configuration by applying a particular operation that consists of two steps:

1. Remove a stick.
2. Add a new stick in such a way that we obtain a new triangulation.

We characterize the operation with an ordered pair of unordered pairs $((a, b), (c, d))$ which signifies that little Fran removed a stick touching vertices a and b, and added a stick touching vertices c and d.

Fran loves hand fans so, while doing these operations, he sometimes asks himself: *"How many operations is needed to transform this triangulation into a "fan" triangulation in vertex x, and, in how many ways is this achievable?".*

Since he is busy doing operations and having fun, he asks for your help!

"Fan" triangulation in vertex x is a triangulation where all diagonals have a common endpoint, namely vertex x.

Let the number of needed operations be m. Let $f_1, f_2, \ldots, f_m$ be a sequence of operations that, when applied in given order, achieves wanted triangulation, thus representing one way of getting there. Let $s_1, s_2, \ldots, s_m$ be another such sequence. Two sequences are distinct if there exists an index i such that $f_i \neq s_i$.

As the number of such sequences can be huge, little Fran is only interested in its remainder modulo $10^9 + 7$.

Input

In the first line are integeres n and q ($4 \leq n \leq 2 \cdot 10^5, 1 \leq q \leq 2 \cdot 10^5$), the number of vertices and the number of events

In each of the next $n-3$ lines there are integers x_i, y_i ($1 \leq x_i, y_i \leq n$), the labels of vertices that the i-th stick touches.

In each of the next q lines there is the integer t_i ($1 \leq t_i \leq 2$) that represents the type of event.

If $t_i = 1$, it is followed by 4 integers a_i, b_i, c_i, d_i ($1 \leq a_i, b_i, c_i, d_i \leq n$) that signify an operation $((a_i, b_i), (c_i, d_i))$ is being made at that moment. It is guaranteed that given operation can be realized.

If $t_i = 2$, it is followed by an integer x_i ($1 \leq x_i \leq n$), which means that little Fran is interested in data for the "fan" triangulation at vertex x_i in relation to the current triangulation.

Output

For every event of type 2, in order they came in input, output two integers, minimal number of operations needed and number of ways to get to the target triangulation using minimal number of operations.

Scoring

Subtask	Points	Constraints
1	12	$n \leq 9, q \leq 1\,000$
2	16	$x_i = 1, y_i = i + 2$ for all $i = 1, \ldots, n - 3$ and there are no events of type 1.
3	30	$q = 1$
4	12	$n, q \leq 1\,000$
5	40	No additional constraints.

Examples

input

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

output

```
0 1
1 1
```

input

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

output

```
1 1
2 1
1 1
```

input

```
9 3
1 5
1 7
2 4
2 5
5 7
7 9
2 1
1 2 5 1 4
2 1
```

output

```
4 12
3 6
```

Clarification of the first example:
Starting triangulation is already a "fan" triangulation in vertex 1, so little Fran must do no operations, which there is one way of doing so. After executing a given operation, there is now only one way to get it to the previous state and that is by applying operation $((2, 4), (1, 3))$.

Clarification of the second example:
Only sequence of operations for the first query: $((3, 5), (1, 4))$.
Only sequence of operations for the second query: $((1, 3), (2, 5)), ((3, 5), (2, 4))$.
Only sequence of operations for the third query: $((3, 5), (2, 5))$.

Task Putovanje

Mr. Malnar has finally reached his annual vacation. The country he decided to travel to can be represented as n cities and m bidirectional roads connecting them. Each road has the same length, and it is possible to reach any city from any other by traveling on these roads. A path from city a to city b is defined as a sequence of roads such that, starting from city a and sequentially traversing the roads in that sequence, one ends up in city b. The length of a path is defined as the number of roads on that path.

Mr. Malnar routinely booked the most expensive hotel in one of the cities and then started to plan his journey. To facilitate his planning, he recorded the length of the shortest path needed from the hotel to each city.

Excited about his long-awaited vacation, Mr. Malnar completely forgot in which city the hotel is located. He certainly does not want to miss the trip, so he asks you to determine in which cities the hotel can be located.

Input

In the first line, there are natural numbers n and m - the number of cities and the number of roads connecting them ($1 \le n \le 5 \cdot 10^4$, $n - 1 \le m \le 10^5$).

In the i-th of the following m lines, there are numbers u_i and v_i - there is a road between cities u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$). There is at most one road between any two cities.

In the last line, there are n integers - the i-th number d_i indicates the distance from the i-th city to the city where the hotel is located, or -1 if Mr. Malnar did not record that distance ($-1 \le d_i < n$).

Output

In the first line, write the number of cities where the hotel can be located.

In the second line, write the labels of the cities where the hotel can be located, **in ascending order**.

Scoring

Subtask	Points	Constraints
1	10	$m + 1 = n \le 5000$, $u_i + 1 = v_i$ for every i
2	20	$d_i = -1$ for every $i > 1$
3	35	$n, m \le 5000$
4	45	No additional constraints.

Examples

input

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

output

```
2
4 6
```

input

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

output

```
2
3 5
```

input

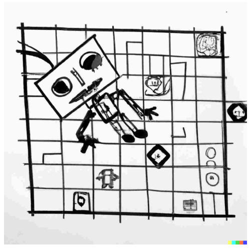
```
4 3
1 2
2 3
3 4
1 -1 -1 1
```

output

```
0
```

Clarification of the first example:
The path from the city labeled 4 to the city labeled 1 is of length 2, while the path from the city labeled 4 to the city labeled 7 is of length 3. Therefore, city 4 satisfies both conditions and the hotel can be located there.
The same holds true for the city labeled 6.

Task Roboti



Kile, a board games enthusiast, recently discovered the game Robots. The game consists of a board with n rows and m columns and one robot. The field $(1, 1)$ is the top-left field of the board, while the field (n, m) is the bottom-right.

At the beginning, the robot is positioned on some field (x, y) (x-th row, y-th column), and the player can direct it in one of the four directions: up, down, left, or right. Depending on the chosen direction, it will move in that direction until it encounters its goal or a special field on the board. If at any point it wants to exit the board, it wraps around to the other side. For example, if it is located at the field $(n, 3)$ and wants to move down, it will arrive at the field $(1, 3)$.

The board has three types of fields:

- Empty field - the robot continues moving in the same direction
- Left turn field - when the robot steps on this field, it will turn left by 90° and continue moving
- Right turn field - when the robot steps on this field, it will turn right by 90° and continue moving

Most fields on the board are empty, only k of them are left or right turn fields.

The game consists of q rounds. In the i-th round of the game, the robot will be placed on the field (a_i, b_i). The goal is to reach the field (c_i, d_i) using the minimum number of turns, or determine that it is impossible.

After playing this game several times, Kile realized that it is more challenging than it initially seemed. That is why he needs your help now. Help him determine the minimum number of turns required for each round of the game!

Note: If the robot starts or finishes its path on a left or right turn field, that turn is not counted.

Input

The first line contains integers n, m and k ($1 \le n, m \le 10^6, 0 \le k \le 10^5$), the number of rows, columns and non-empty fields on the board.

The i-th of the following n lines contains integers x_i, y_i and character s_i ($1 \le x_i \le n, 1 \le y_i \le m, s_i =$ `'L'` $or s_i =$ `'R'`), the row and column of i-th turn field and the type of turn. If $s_i =$ `'L'` then it is a left turn field. If $s_i =$ `'R'` then it is a right turn field.

The next line contains integer q ($1 \le q \le 3 \cdot 10^5$), the number of rounds.

The i-th of the following q lines contains integers $a_i, b_i, c_i, d_i (1 \le a_i, c_i \le n, 1 \le b_i, d_i \le m)$, the starting position and the goal.

Output

In the i-th of the following q lines output the minimal number of turns for the i-th round of the game or `'-1'` if it is impossible to reach the goal.

Scoring

Subtask	Points	Constraints
1	10	$k = 0$
2	13	$n, m \le 300, q \le 10$
3	49	$n, m \le 300$
4	38	No additional constraints.

Examples

input

```
2 2 2
1 1 L
2 2 R
5
1 1 2 2
2 1 1 2
1 1 1 2
2 1 1 1
2 2 2 1
```

output

```
-1
1
0
0
0
```

input

```
3 3 4
1 1 L
1 3 L
2 1 L
3 3 L
7
1 1 3 3
3 3 2 1
3 1 2 2
2 3 1 2
2 3 3 1
1 2 3 2
3 3 2 2
```

output

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

input

```
4 4 8
1 1 R
1 3 L
2 2 R
2 4 L
3 1 L
3 3 L
4 2 L
4 4 L
7
1 2 1 4
2 2 3 4
4 4 3 2
4 1 4 4
4 2 3 1
1 2 2 3
2 4 2 3
```

output

```
2
1
1
0
-1
5
0
```

Clarification of the second example:
First round: We start at (1, 1). If we direct the robot to the left, it will arrive at (1, 3) in the next step because it wanted to exit the board, so it wraps around to the other side. Field (1, 3) is a left turn field, so the robot is now directed downwards. After two more steps, it will be at the desired goal (3, 3).

Second round: We start at (3, 3). If we direct the robot upwards, it will arrive at (1, 3) in two steps, where it will be directed to the left due to the left turn field. After two steps, it will be at the field (1, 1), which is also a left turn field, so it will be directed downwards. In the next step, it will be at the desired goal (2, 1).