Toy

For creating a task for CEOI 2024, Ben was given a toy as a present from the scientific committee. The toy is a puzzle which can be imagined as a $H \times W$ grid containing a metal object consisting of two parts: A horizontal $1 \times K$ part and a vertical $L \times 1$ part, which are loosely attached to each other. Neither of the parts can be rotated in any way, but each can be slid vertically or horizontally independently of the other one, as long as they always overlap on exactly one square.

Furthermore, the grid contains several obstacles. No part of the metal object can move through an obstacle. Worse yet, the parts also cannot move outside the grid, not even partially. Ben's task is to move the metal object from a designated starting location to a (possibly) different location so that both parts overlap on a designated target square.

However, Ben has been playing with the toy for a while and he has not yet been able to solve the task. In fact, he has gained a suspicion that the organizers have played a prank on him and have given him an unsolvable puzzle. He thus asks for your help by telling him whether the puzzle is solvable or not.

Input

The first line of the input contains four space-separated integers W, H, K and L — the width and the height of the puzzle, the width of the horizontal part and the height of the vertical part, respectively. The second line contains four integers x_h, y_h, x_v and y_v — the coordinates of the leftmost square occupied by the horizontal part and the coordinates of the topmost square occupied by the vertical part.

The rows are numbered from 0 to $H-1$ from top to bottom and columns are numbered 0 to $W-1$ from left to right. The x coordinate denotes the column number and y coordinate denotes the row number.

The next H lines contain W characters each, representing the grid. The character `.` represents an empty square, the character `X` represents an obstacle and the character `*` represents the target square.

It is guaranteed that the initial position of the metal object is valid, i.e., that the two parts overlap on exactly one square and the two parts neither overlap with an obstacle nor stick out from the grid.

There is a single target square, i.e., a single occurrence of the `*` symbol in the toy, which might overlap with the initial position of the metal object.

Output

Print a single line containing `YES` if it is possible to move the metal object to the target square, `NO` otherwise.

Examples

Example 1

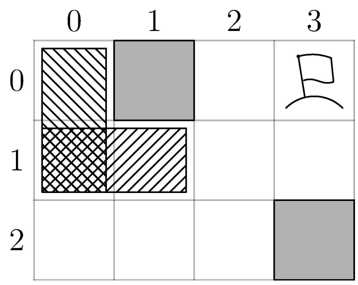
Input:

```
4 3 2 2
0 1 0 0
.X.*
....
...X
```

Output:

```
YES
```

The initial situation looks as follows:



We can reach the target square by first moving the vertical part one square down, followed by alternating movement to the right of the vertical and horizontal parts as long as possible. Then we can move the vertical part up and to the right, reaching the target square, and finally move the horizontal part up, reaching the target as well.

Example 2

Input:

```
2 3 2 3
0 1 0 0
.X
.*
.X
```

Output:

```
NO
```

There is no way how to move the vertical part without running into an obstacle. Therefore, it can never reach the target square.

Constraints

- $2 \leq W, H \leq 1\,500$
- $2 \leq K \leq W$, $2 \leq L \leq H$
- $0 \leq x_h \leq W - K$, $0 \leq y_h \leq H - 1$
- $0 \leq x_v \leq W - 1$, $0 \leq y_v \leq H - L$

Subtasks

1. (14 points) $W, H \leq 50$
2. (21 points) $W, H \leq 90$
3. (9 points) $W, H \leq 300$ and $K, L \leq 10$
4. (29 points) $W, H \leq 360$
5. (27 points) *no additional constraints*

Petrol stations

The Czech highway network consists of N cities and $N-1$ roads with known lengths in kilometers. We know that there exists exactly one path between each pair of cities. Furthermore, there exists exactly one petrol station in each of the cities and nowhere else.

One day, several people decided to go on a car trip. There were a total of N^2 cars traveling. Strangely, it holds that for each ordered pair of cities (a, b) there was exactly one car going from the city a to the city b, traveling alongside the only path between these cities. Since everyone in Czechia uses Škoda cars, every car has the same fuel tank capacity of K liters and they steadily consume one liter of petrol per kilometer traveled. Before departure, the fuel tank of each car is full. Furthermore, the Czechs are quite predictable. Due to their laziness, they refuel only when they don't have enough petrol to reach the next city (entering a city with an empty tank is still possible). Once they are forced to stop at a petrol station, they always fill their tank fully.

The Czech tax authority would like to know how many cars stopped at each petrol station during the day. Given this predictable behavior, you should be able to compute it easily.

Input

The first line of the input contains two space-separated integers N and K — the number of cities and the capacity of the fuel tank of each car. The following $N-1$ lines describe roads. Each of them contains three space-separated integers u_i, v_i and l_i, where u_i and v_i are indices of cities connected by the i-th road and l_i is the length of this road in kilometers. Cities are numbered from 0 to $N-1$. It is guaranteed that for every pair of cities, there exists exactly one path between them.

Output

You should output N lines, which should contain the number of cars stopping at the petrol station in each city, ordered from city 0 to city $N-1$.

Examples

Example 1

Input:

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

Output:

```
0
2
0
```

There are three cites in a line connected by roads of length 1 and the fuel tank has a capacity of 1 liter. Only the cars going between the two outer cities will stop in the middle city.

Example 2

Input:

```
6 2
0 1 1
1 2 1
2 3 1
3 4 2
4 5 1
```

Output:

```
0
3
3
12
8
0
```

This time there are 6 cities in a line and the fuel tank has a capacity of 2 liters. Many cars need to stop in cities 3 and 4. This makes sense because cities 3 and 4 are connected by a road with a length of 2 kilometers.

Constraints

- $2 \le N \le 70\,000$
- $1 \le K \le 10^9$
- $0 \le l_i \le K$ (for each i such that $0 \le i \le N - 2$)

Subtasks

Let D denote the maximum number of roads connected to a single city.

1. (18 points) $N \le 1\,000, K \le 1\,000$
2. (8 points) $D \le 2$ and $l_i = 1$ (for each i such that $0 \le i \le N - 2$)
3. (10 points) $D \le 2$
4. (12 points) $K \le 10, D \le 10$
5. (17 points) $K \le 10$
6. (35 points) *no additional constraints*

Sprinklers

Václav has a beautiful flower garden consisting of M flowers planted on a single line. On this line, Václav has also placed N sprinklers to water his flowers.

The positions of the sprinklers are given by the numbers $s_1, \ldots, s_N$. The positions of the flowers are given by the numbers $f_1, \ldots, f_M$. Both are provided in non-decreasing order, that is:

- $s_1 \leq s_2 \leq \ldots \leq s_N$
- $f_1 \leq f_2 \leq \ldots \leq f_M$

Václav is leaving for CEOI soon. He would like to make sure that all of his flowers are properly watered while he is away. To do this, he turns each sprinkler individually to the left or to the right, and sets their spraying power — all sprinklers share the same water hose, and therefore spray the same distance.

If the spraying power is K and the i-th sprinkler is turned to the left, it will water all flowers with positions between $s_i - K$ and s_i (inclusive). Similarly, if the j-th sprinkler is turned to the right, it will water all flowers with positions between s_j and $s_j + K$ (inclusive). A single sprinkler can water multiple flowers and a single flower can be watered by multiple sprinklers.

Your task is to decide whether it's possible to water all the flowers. If so, you should find the minimum sufficient spraying power, along with a corresponding configuration of sprinklers. If there exist multiple valid configurations with minimal spraying power, output any of them.

Input

The first line of input contains two integers: N and M, separated by a space. The second line contains N space-separated integers $s_1, \ldots, s_N$ — the positions of the sprinklers. The third line contains M space-separated integers $f_1, \ldots, f_M$ — the positions of the flowers.

Output

If it is not possible to water all the flowers, print the number -1.

If it is possible, the output should consist of two lines. On the first line, output the number K – the minimum spraying power required to water all the flowers. On the second line, print a string c of length N, such that c_i is `L` if the i-th sprinkler should be turned to the left and `R` otherwise.

Examples

Example 1

Input:

```
3 3
10 10 10
5 11 16
```

Output:

```
6
LLR
```

The given solution is valid — each flower is watered by at least one sprinkler. A spraying power lower than 6 is not possible, because the flower at location 16 is 6 units away from the closest sprinkler.

Example 2

Input:

```
1 2
1000
1 2000
```

Output:

```
-1
```

At most one flower can be watered at one time regardless of the orientation of the only sprinkler.

Constraints

- $1 \leq N, M \leq 10^5$
- $0 \leq s_i \leq 10^9$ (for each i such that $1 \leq i \leq N$)
- $0 \leq f_i \leq 10^9$ (for each i such that $1 \leq i \leq M$)
- $s_i \leq s_j$ for all $i \leq j$
- $f_i \leq f_j$ for all $i \leq j$

Subtasks

1. (3 points) $N = 1$
2. (6 points) $N = 3x$ for some integer x, $s_{3i+1} = s_{3i+2} = s_{3i+3}$ for each i such that $0 \le i \le x - 1$ (i.e. sprinklers are always placed in groups of three)
3. (17 points) $N \le 10, M \le 1\,000$
4. (27 points) $K \le 8$ (i.e., in all testcases there exists a configuration of sprinklers such that a spraying power of at most 8 is sufficient to water all of the flowers)
5. (47 points) *no additional constraints*