CEOI 2022

CENTRAL EUROPEAN OLYMPIAD IN INFORMATICS

VARAZDIN, CROATIA, JULY 24 - 30

Day 2

July 28th 2022

Tasks

Task	Time Limit	Memory Limit	Score
Drawing	1.5 seconds	512 MiB	100
Measures	1.5 seconds	512 MiB	100
Parking	2 seconds	512 MiB	100
Total			300

REPUBLIC OF CROATIA
Ministry of Science and Education

CROATIAN ASSOCIATION OF TECHNICAL CULTURE

CROATIAN COMPUTER SCIENCE ASSOCIATION

Task: Drawing

Paint & Wine is the first painting studio in Zagreb offering relaxing painting lessons with a glass of wine on the side. During the lesson, students are given a certain theme, and with the aid of master painters usually manage to paint an impressive piece.

Ante is a master painter, Luka is his student, and this task tells the tale of a lesson that included a bit more wine than usual.

Ante: "Paint me a tree!"

Luka: "Alright. What kind of tree do you want? Palm, oak, pine. . . ?"

Ante: "I want a connected acyclic undirected graph!"

Luka: "I can do that. . . Any other wishes?"

Ante: "I like it when no node is adjacent to more than three other nodes!"

Luka: "Uhm, okay. . . Well, there are many such trees."

Ante: "Here is a list of edges, I want that one!"

Luka: "Ok, wow. Still, there are many ways to draw it."

Ante: "Here is a list of points in the plane where I want the nodes to be drawn. I also don't want to see a pair of intersecting edges."

Luka: "I'm on it!"

Your task is to help Luka paint the tree according to Ante's wishes. More precisely, given a description of a tree, such that no node is adjacent to more than three other nodes, and a list of points in the plane, find a one-to-one mapping of nodes to points such that, when edges of the tree are drawn as line segments between corresponding points, they do not intersect (except at end points).

Input

The first line of input contains an integer N, the number of nodes in the tree and the number of points in the plane.

The following $N - 1$ lines describe edges of the tree, one per line. Each edge is described by two integers a and b, labels of nodes connected by the edge. Nodes are labelled with integers from 1 to N.

It is guaranteed that each node is adjacent to at most three other nodes.

The following N lines contain the points to be used when drawing the tree, one per line. Each point is described by a pair of integer coordinates. No two points share the same pair of coordinates, and **no three points lie on the same line**.

Output

Output a permutation of integers from 1 to N in a single line. The i-th number should be the label of the node which is mapped to the i-th input point.

If there are multiple valid solutions, output any of them. It's guaranteed that a solution always exists.

Scoring

In all subtasks point coordinates are integers between 0 and 10^9.

Subtask	Score	Constraints
1	10	$3 \leq N \leq 200\,000$, there exists a convex polygon with the given points as vertices
2	15	$1 \leq N \leq 4\,000$
3	15	$1 \leq N \leq 10\,000$
4	35	$1 \leq N \leq 80\,000$
5	25	$1 \leq N \leq 200\,000$

Examples

input

```
3
1 2
2 3
10 10
10 20
20 10
```

output

```
1 2 3
```

input

```
5
1 2
1 3
1 4
4 5
10 10
10 30
30 10
30 30
20 25
```

output

```
5 4 2 3 1
```

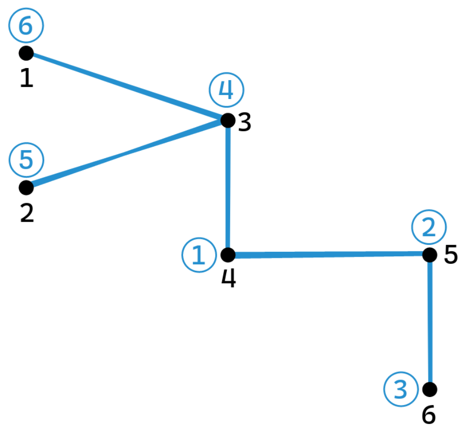
input

```
6
1 2
2 3
1 4
4 5
4 6
10 60
10 40
40 50
40 30
70 30
70 10
```

output

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

Clarification of the third example:



Blue numbers represent node labels while black numbers represent point indices.

Task: Measures

The COVID-19 pandemic took the world by surprise in many ways. Almost overnight, people around the globe had to adapt to a new way of life, mainly shaped by the preventive measures issued by their local authorities, all with the goal of suppressing and controlling the spread of the disease.

To better prepare for the unlikely event of a more devastating outbreak in the far future, the Croatian National Institute of Public Health decided to open various research departments. The main goal of these departments is to develop highly efficient protocols that help the general population to quickly adhere to a new preventive measure.

Alenka works in one such department, and is currently investigating the scenario in which a group of people stands in a line, e.g. in front of a post office, and suddenly a new safety measure takes place, mandating that distance between any two people has to be at least D.

She also implemented an app that allows the user to specify a distance D and locations of N people as coordinates along a line. The app then draws a picture of a line which represents the situation, and calculates the smallest amount of time in seconds, denoted as t_{opt}, needed for the group to reach a new arrangement that satisfies the preventive measure. The app assumes that people are going to immediately start rearranging themselves optimally, and that all people move with the same constant speed of one unit per second.

She now wants to add a new feature that will enable the user to add M additional people to the group by tapping on the drawn line, thereby specifying their locations. The app is supposed to recalculate t_{opt} after each tap, i.e. after each new person is added to the group.

Your task is to help Alenka by implementing this feature.

Input

The first line contains integers N, M, and D from the task description.

The second line contains N integers $a_1, \ldots, a_N$, the locations of the N initial people.

The third line contains M integers $b_1, \ldots, b_M$, the locations of the M additional people.

Output

Output M numbers in one line, the i-th of them representing the value of t_{opt} given that the group consists of $(N+i)$ people at locations a_1, a_2, $\ldots$, a_N, b_1, $\ldots$, b_i.

Output each number in decimal notation without trailing zeroes, e.g. output `1.23` instead of `1.2300`, and `123` instead of `123.` or `123.0`. It can be proven that answers always have a finite decimal representation.

Scoring

In all subtasks it holds that $1 \leq D, a_1, \ldots, a_N, b_1, \ldots, b_M \leq 10^9$.

Subtask	Score	Constraints
1	10	$0 \leq N \leq 2\,000$, $1 \leq M \leq 10$
2	14	$0 \leq N \leq 200\,000$, $1 \leq M \leq 10$
3	35	$N = 0$, $1 \leq M \leq 200\,000$, $b_1 \leq \cdots \leq b_M$
4	41	$N = 0$, $1 \leq M \leq 200\,000$

Examples

input

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

output

```
1
```

input

```
0 5 3

1 2 3 4 5
```

output

```
0 1 2 3 4
```

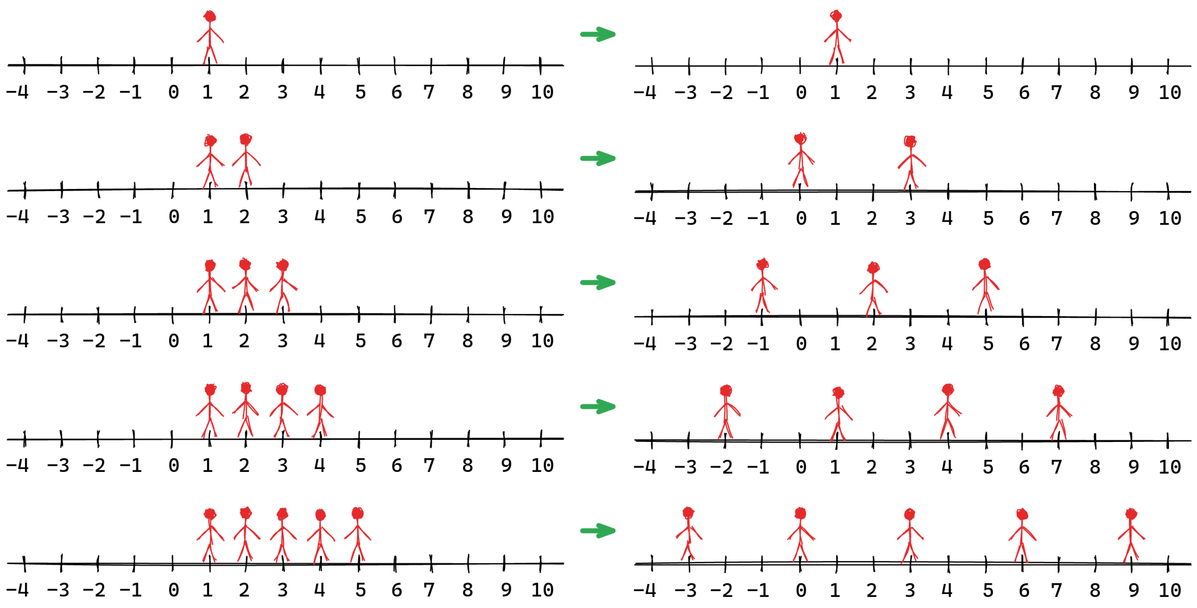
input

```
3 3 3
3 3 3
3 3 3
```

output

```
4.5 6 7.5
```

Clarification of the second example:



Task: Parking

Valerija works as a valet at a fancy restaurant. Her job is to wait for the arrival of distinguished guests, politely greet them, obtain the keys of their vehicles, and park their vehicles at a nearby parking lot. Once the event is over, she makes sure that each of the guests regains custody of their vehicle and happily leaves the venue.

One evening, shortly after she finished parking all of the vehicles, she noticed a particularly interesting property regarding their colors. Namely, it turned out there were exactly $2N$ vehicles at the parking lot, and they were colored in N different colors, such that there were exactly two vehicles of each color. We denote the colors of the vehicles by integers from 1 to N.

The parking lot itself is organized as a sequence of M parking spaces, denoted by integers from 1 to M, where each parking space can contain at most two vehicles. There is only one entrance to a parking space, and a vehicle can enter or exit the space if no other vehicle is blocking the entrance. We'll call the vehicle parked closer to the entrance the *top vehicle*, and the vehicle parked further from the entrance the *bottom vehicle*. Valerija parked the vehicles in such a way that each parking space is either empty, full (i.e. contains two vehicles), or contains a single bottom vehicle.

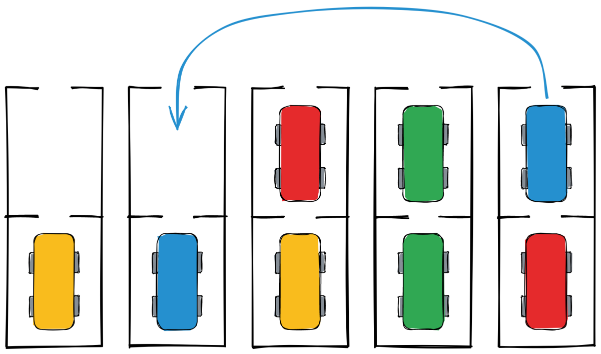


Illustration of the first example, showing the only possible first drive.

Valerija would like to repark the vehicles so that each pair of vehicles of the same color is parked in the same parking space. She doesn't care which parking space will contain which color, and which specific vehicle will be at the top or bottom of the parking space. She will repark the vehicles in a series of *drives*. In each drive, she will sit in a parked vehicle that is able to exit its current parking space, and will drive it to another parking space that is either:

- empty, in which case she parks it as a bottom vehicle, or
- contains a single parked vehicle **of the same color** as the one she's currently driving, in which case she parks it as a top vehicle.

Valerija wants to minimize the number of drives needed to repark the vehicles according to her wishes. Your task is to help her by finding the shortest sequence of drives that would achieve her goal, or determine that no such sequence exists.

Input

The first line contains two space-separated integers N and M from the task description.

The i-th of the next M lines contains two space-separated integers b_i and t_i ($0 \leq b_i, t_i \leq N$) that describe the i-th parking space. More precisely, the number b_i represents the bottom vehicle color, and the number t_i represents the top vehicle color. If a position in the parking space is empty, the corresponding integer will be equal to 0. It is guaranteed that no parking space contains only a top vehicle, i.e. if $b_i = 0$, then $t_i = 0$ as well.

Output

If there is no sequence of drives that could repark the vehicles according to Valerija's wishes, print -1 in the only line.

Otherwise, the first line should contain an integer K, the smallest number of drives needed to satisfy Valerija's goal.

The i-th of the next K lines should describe the i-th drive. More precisely, it should contain two integers, x_i and y_i ($1 \leq x_i, y_i \leq M, x_i \neq y_i$), representing that Valerija should take a vehicle from parking space x_i to parking space y_i in the i-th drive. Of course, the x_i-th parking space must contain at least one vehicle at that point, and the vehicle closer to the entrance must be movable to parking space y_i, i.e. parking space y_i must be either empty or contain a vehicle of the same color.

Scoring

In all subtasks it holds that $1 \leq N \leq M \leq 200\,000$.

If your solution correctly determines the smallest number of drives in all test cases of a certain subtask, but incorrectly outputs the description of drives in some of them (or doesn't output it at all), it will receive 20% of the points allocated to that particular subtask.

Subtask	Score	Constraints
1	10	$M \leq 4$
2	10	$2N \leq M$
3	25	All parking spaces are initially either empty or full, and $N \leq 1\,000$.
4	15	All parking spaces are initially either empty or full.
5	25	$N \leq 1\,000$
6	15	No additional constraints.

Examples

input

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

output

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

input

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

output

```
-1
```

input

```
5 7
1 0
2 1
2 3
4 3
5 4
5 0
0 0
```

output

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

Clarification of the first example: The image from the task description depicts the initial state of the parking lot in this example. Notice that in this case, each drive is forced, i.e. there is only one valid first drive, only one valid second drive, and two equivalent third drives, after which we reach the end goal.