Maratona Inter-Universitária de Programação

7th of October - 10:00 to 15:00

Universidade do Minho

accenture
High performance. Delivered.

KEEPSOLUTIONS
University of Minho SPIN-OFF

Acknowlegments

cesium

NECC

Benedita Henriques

Contents

Scientific Committee

- Alberto Simões (Instituto Politécnico do Cávado e Ave)
- André Restivo (Faculdade de Engenharia, Universidade do Porto)
- Fábio Marques (Escola Superior de Tecnologia e Gestão de Águeda, Universidade de Aveiro)
- Filipe Araújo (Faculdade de Ciências e Tecnologia, Universidade de Coimbra)
- Hugo Ferreira (Faculdade de Engenharia, Universidade do Porto)
- José Saias (Escola de Ciências e Tecnologia, Universidade de Évora)
- Margarida Mamede (Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa)
- Paul Crocker (Faculdade de Engenharia, Universidade da Beira Interior)
- Pedro Guerreiro (Faculdade de Ciências e Tecnologia, Universidade do Algarve)
- Pedro Mariano (Faculdade de Ciências, Universidade de Lisboa)
- Pedro Ribeiro (Faculdade de Ciências, Universidade do Porto)
- Rui Mendes (Departamento de Informática, Universidade do Minho)
- Simão Sousa (Faculdade de Engenharia, Universidade da Beira Interior)

Local Organization Committee

- Pedro Henriques
- Solange Lima
- Goretti Pereira
- Pedro Nuno Sousa
- Manuel Alcino
- Víctor Fonte
- J.João Almeida
- Rui Mendes

Compilers

Lang.	Version	compile	execute
C	gcc $\geq$ 5.4	gcc -g -O2 -Wall -Wextra -std=gnu11 -static *name.c* -lm	*a.out*
C++	g++ $\geq$ 5.4	g++ -g -O2 -Wall -Wextra -std=gnu++14 -static *name.cpp* -lm	*a.out*
Java	javac $\geq$ 8	javac *name.java*	*java* name

Compilation Constraints

- **Maximum compilation time:** 60 seconds
- **Maximum source code size:** 100 KB
- Every source code must be submitted in a single file
- In case of Java submissions, the `.java` file has to have the same name as the class that contains the main method. There is no limit for the number of classes to be contained in that file.

Runtime Constraints

These limits depend on the problems.

- **Maximum CPU time:**([AGI]:2, B:4, C:5, [DEH]:1, F:3) second
- **Maximum Memory:** 256MB

About the Input/Output

- All lines (both in the input and output) should be ended by the newline character (`'\n'`)
- Except when explicitly stated, single spaces is used as a separator.
- No line starts or ends with any kind of whitespace.

Documentation

Language documentation is available, namely:

- man pages for C/C++ function *(using the command line)*
- C++ STL documentation *(bookmarked on your browser; link on Mooshak first page)*
- Java SE Documentation *(bookmarked on your browser; link on Mooshak first page)*

Problem: The Suspicious Backpack

Final-year students from the Police Academy were handed a challenging team project as part of their training. A suspicious backpack was found in a huge shopping centre and they must identify the person who put it there. To make things harder for them, the camera surveillance system had been switched off. Thus, police trainees could only interview witnesses and have heard of many *suspects* who had been at the gallery where the backpack was left. In the sequel, assume that every suspect has been at the gallery exactly once.

After all the interviews were over, students put together the information they collected, which is of the following two types:

- *Preceding conjecture:* A suspect X was at the gallery before another suspect Y. More precisely, X left the gallery before Y arrived.
- *Concurrent conjecture*: At some moment in time, two different suspects, X and Y, were both at the gallery.

Before proceeding with the investigation, the students want to know if anyone lied. In other words, they want to find out whether the set of all gathered conjectures is *inconsistent*, in the sense that the described scenario is impossible.

Let us look at two examples with four suspects: Anne, Bob, Cora, and Dan. In both examples, the students have learned that Anne was at the gallery before Bob, that Cora was at the gallery before Dan, and that Bob and Cora were both at the gallery at some moment.

- First, suppose they have also learned that Bob and Dan were at the gallery at the same time. All this is possible (the set of conjectures is consistent). For a proof, consider the following sequence of events which is compatible with all conjectures: Anne arrived, Anne left, Bob arrived, Cora arrived, Cora left, Dan arrived, Bob left, Dan left.
- Now suppose that, instead of Bob and Dan, it is Anne and Dan who were at the gallery at the same time. In this case, someone must have lied (the set of conjectures is inconsistent). As every suspect has been at the gallery only once, "Anne before Bob" and "Bob and Cora at some moment" imply that Anne left before Cora left. Since Cora left before Dan arrived, Anne and Dan could not have been simultaneously at the gallery.

Task

Write a program that, given a set of suspects, a set of preceding conjectures and a set of concurrent conjectures, finds out whether the set of all conjectures is inconsistent.

Input

The first line has three integers: S, which is the number of suspects, P, which is the number of preceding conjectures, and C, which is the number of concurrent conjectures. Suspects are identified by integers, ranging from 0 to $S - 1$.

Each of the following P lines specifies a different preceding conjecture. It has two distinct integers, x and y, which indicate that suspect x was at the gallery before suspect y.

Each of the following C lines specifies a different concurrent conjecture. It has two distinct integers, x and y, which indicate that, at some moment, suspects x and y were both at the gallery.

Constraints

$2 \leq S \leq 20\,000$	Number of suspects
$1 \leq P \leq 50\,000$	Number of preceding conjectures
$1 \leq C \leq 50\,000$	Number of concurrent conjectures
$0 \leq x, y < S$	Suspects

Output

The output has one line with: `Inconsistent conjectures`, if the set of all conjectures is inconsistent; `Consistent conjectures`, otherwise.

Sample Input 1

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

Sample Output 1

```
Consistent conjectures
```

Sample Input 2

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

Sample Output 2

```
Inconsistent conjectures
```

Sample Input 3

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

Sample Output 3

```
Consistent conjectures
```

Problem: The Parades

The sun is beating down as you look down from your VIP stand after a very nice lunch and observe the colourful spectacular below, a sea of humanity in an array of brightly colourful costumes. It's almost time for the traditional North Versus South Parade where the most respected members from each region will line up and parade past the public and VIP stands. A voice behind you interrupts your thoughts, "It's almost time, the master of ceremonies wants you to calculate and announce the winners". The winners from each side will form an identically coloured sequence and then parade together receiving great honour and tribute from the two presidents. A certain unease descends upon you as you realize the truth of the old saying, there is no such thing as a free lunch.

Task

Announcing the winners is a complicated and difficult task, when the signal is given the members in the parades have to form up into two lines in what to many seems like some seemingly random sequence and then begin the parade. The winners are the parade members that from the longest common coloured subsequence contained in each line. In this case to avoid ties we always want the lexicographically smaller sequence – in other words blue,green,red is better than red, green, blue. As there are a lot of colours we shall restrict ourselves to 52 colours and label them A..Za. . . z. Your task is to calculate the number of winners from each side and the sequence of colours.

Input

Two lines of input. Each line will contain one string of at most 98 characters.

Output

Two lines of output (terminated with a new line character). On the first line an integer with the length of the winning sequence and on the second the actual sequence (the lexicographically smaller in case of a tie). In the case where there is no common sequence then the output is only one line with the value zero (0).

Sample Input 1

```
abcd
fbcf
```

Sample Output 1

```
2
bc
```

Sample Input 2

```
CabD
CDab
```

Sample Output 2

```
3
Cab
```

Sample Input 3

```
xxabABxxcdCDefEF
ABxyabCDxycdEFef
```

Sample Output 3

```
8
ABxxcdEF
```

Sample Input 4

```
wjfljrglejrfrlfjejflerjflerjflejwwwwaaaswswxxwqq
ljrrjfglejrfljrlfjelrjflerjflerjflejwwwwwxxwqq
```

Sample Output 4

```
39
ljrglejrfrlfjejflerjflerjflejwwwwwxxwqq
```

Problem: Please, sit down, we need to negotiate

Negotiations are a black magic art. Before negotiating and having failed to have everyone involved around the table, we need to gather the right people with the right mindset. Without that, the negotiations are doomed to fail.

But how to ensure that, when no one wants to join the negotiation without some established predefined conditions?

Fortunately, here, these conditions are simply stated: "I join the negotiation table if X and Y and ... and Z also agreed to join"

If we number incrementally each potential participant by a natural number, say from 0 to $n-1$, it is important to ensure that j, the key participant for the negotiation, join it, knowing that each potential participant has a certain number of condition to join the table.

Gathering the people around the table is done in rounds, iteratively. One negotiator joins the table if the people already seated meet his own requirement. By doing that, other negotiators may join the table the next round, because of him.

Task

Taking into account which participant is the key in the negotiation process, and all the negotiator's constraints, your task is to tell the head of the negotiation process that the right conditions are met for a fruitful negotiation.

Input

The first line introduces two numbers: n and m separated by a single space.
The number n is the number of potential negotiators (numbered from 0 to $n-1$) and m is the number of overall stated conditions.

The second line introduces the id (i.e. the number) of the key negotiator.
The following m lines introduce the conditions with the following format:

$r\ p_1\ \ldots\ p_r\ p$ (with $0 \leq r < n$)

and with the meaning: if p_1 and $\ldots$ p_r join the table, then p also joins the table

Hints The condition 0 p means that p is willing to join the table without any conditions. A potential negotiator with no stated condition is a negotiator that categorically refuses to join the negotiation table.
If a potential negotiator has more than one stated condition, this means that he is willing to join the table if at least one of his conditions is fulfilled.

Constraints

Potential participating members: $2 \leq n \leq 5000$
Number of stated conditions to join the negotiation table: $0 < m \leq 5000$

Output

A single line with: "YES" if the key negotiator is able to join the negotiation, "NO" otherwise.

Sample Input 1

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

Sample Output 1

```
NO
```

Sample Input 2

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

Sample Output 2

```
YES
```

Problem: The Casino Hunters

Nick and his friends have a long history of hijacking slot machines in casinos. They managed to buy one of these machines, and discovered that they use a simple linear congruential generator inside, like

$$x_{i+1} = ax_i + b \pmod{c}$$

to generate the sequence of pseudo-random numbers that determine the figures showing up in the screen. As soon as they learn the internal state of the machine from the figures they see, Nick and his friends can anticipate which figures are coming next, to play at the right time.

Now, the casino hired a famous detective to prove that Nick's team has been fooling the system. However, this famous detective is no longer at his peak and came for your help. This detective is no other than Sir Luck Gomes himself. Could you help him?

Task

You will be given four numbers of the sequence, x_0, x_1, x_2 and x_3. Your task is to output the next number of the sequence, x_4. You should assume the smallest possible (prime) c that is consistent with the inputs.

Input

x_0
x_1
x_2
x_3

Constraints

Constants a, b and c are prime numbers. The following constraints apply:

$3 < c < 30,000$
$0 < a < c$
$0 < b < c$
$a \neq b$
$0 \leq x_0 < c$

Output

x_4

Sample Input 1

```
9
0
7
5
```

Sample Output 1

```
10
```

Sample Input 2

```
7992
3036
13497
12917
```

Sample Output 2

```
4904
```

Problem: Jack-o'-lantern

It's Halloween and Jack hates Halloween; the city is dark, streets are not well-lit and strange creatures linger in every shadow. For that reason, when he goes home at night, he never walks through streets without street lamps.

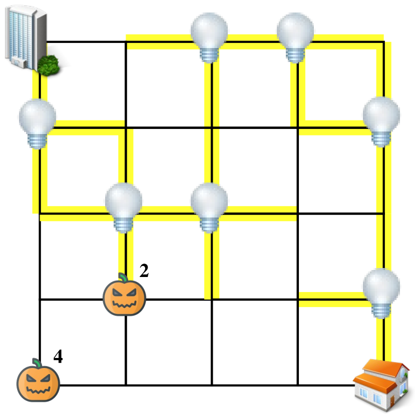
On the other hand, city planners have done a terrific job organizing the city. Every block is a perfect square, all have the same size and all roads are laid out in a grid.

Yet, it is dark, so really dark! Street lamps are only placed at intersections and are often non-functional making the city even scarier.

Jack wants to get home as quickly as possible, but only using well-lit streets. A street is a piece of road that connects two intersections and Jack considers it to be well-lit if at least one of the intersections has a working street lamp.

Luckily, as it is Halloween, lanterns are often found throughout the city, but never near street lamps. Each lantern has a candle that Jack can use to go through a certain number of streets. Jack can put the candle out when not needed and light it up again to go through a dark street.

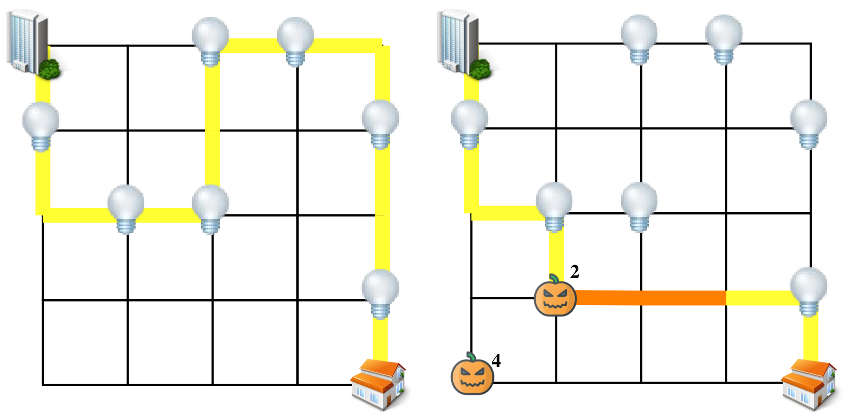
Given the bulkiness of the lanterns, Jack can only carry one with him at any given time. He can, of course, drop it and pick another one. Lanterns are often scary, but not as scary as a gloomy autumn night.



Task

Knowing where working street lamps are placed throughout the city, and the position and the duration of each lantern, find the fastest route that Jack can take from his work to his house.

Considerer that Jack's work is always in the top-left corner of the map, his house is in the bottom-right corner, and there is always a possible solution. The next figure shows possible shortest paths with and without lanterns in the city.



Input

The first line contains two integers (w and h) that represent the width and height of the city.

The next h lines, each contains w characters that each represents an intersection. These characters can be: a star (*) representing an intersection with a street lamp, a zero (0) representing an empty intersection, or an integer (d) representing a lantern that can be used to go through d dark streets.

Constraints

$1 < w \leq 100$	Width of the city.
$1 < h \leq 100$	Height of the city.
$1 \leq d \leq 8$	Duration of a lantern.

Output

A single line containing a single integer representing the length of a shortest path that Jack can take to get home.

Sample Input 1

```
5 5
0 0 * * 0
```

```
* 0 0 0 *
0 * * 0 0
0 0 0 0 *
0 0 0 0 0
```

Sample Output 1

```
12
```

Sample Input 2

```
5 5
0 0 * * 0
* 0 0 0 *
0 * * 0 0
0 2 0 0 *
4 0 0 0 0
```

Sample Output 2

```
8
```

Problem: Invoice Management

Success Business

99 Jefferson Avenue, Suite 99,
Redwood City, CA 94063
Tel: 650-275-7632
Mail: info@succcess_business.com

INVOICE

PAID

TO:

Perfect Suppliers
777 Road Madison,
AL 35758
Telephone: 212-290-4780.

SHIP TO:

World Eco Products
88 5th Street S.W.,
Washington, DC 20410.
Telephone: -203-708-9152

SALESPERSON	P.O. NUMBER	REQUISITIONER	SHIPPED VIA	F.O.B. POINT	TERMS
					Due on receipt

QUANTITY	DESCRIPTION	UNIT PRICE	TOTAL
1	Non-Reflective Mirror	119.95	$119.95
1	Silent Alarm Clock	139.95	$139.95
1	Teflon Bath Mat	7.95	$7.95
5	Water-Resistant Sponge	60.05	$300.25
5	Water-Soluble Dishcloth	60.05	$300.25
3	Black Light Bulb	40.09	$120.27

Jakob is an acountant in a small firm. This firm has lots of business but shoddy management. There are many invoices, most of them of small amounts and lots of payment orders. The problem is that most payment orders do not agree with the invoices. Jakob wants to get rid of as much invoices and payment orders as possible whislt also liquidating as much of the debt as possible.

Your task is to find, for each client, the **largest** amount of A such that:

$$A = \sum_i I_i = \sum_j P_j,$$

using as much invoices, I_i and payments P_j as possible.

Task

Your program should read a file with invoice values (identified by the client ID and total amount) and payments (client ID and credit value).

You should print, for each client, a line with the client id, the maximum number of invoices and the maximum number of payments (sorted by client id).

Input

Input is a tab separated table listing invoices and payments. The first line will include the number of clients, the number of invoices, and the number of payments to be handled (all separated by spaces):

```
3 9 6
```

Invoices have the following syntax (client ID and amount):

```
I 1 7
I 1 3
I 1 4
I 1 1
I 1 2
```

```
I 2 3
I 3 3
I 3 7
I 3 4
```

Payments have the following syntax (client ID and amount):

```
P 1 6
P 1 4
P 1 4
P 2 2
P 2 2
P 3 7
```

Constraints

- client ID is an integer;
- invoices amounts and payment amounts are positive integer values;
- there are no more than 400 clients, 200 invoices and 200 payments per client;
- all accouting can be performed with 32 bits unsigned integers.

Output

For each client id, the number of invoices actually paid, and the number of used payments.

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

Sample Input

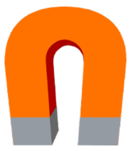
```
3 9 6
I 1 7
I 1 3
I 1 4
I 1 1
I 1 2
I 2 3
I 3 3
I 3 7
I 3 4
```

```
P 1 6
P 1 4
P 1 4
P 2 2
P 2 2
P 3 7
```

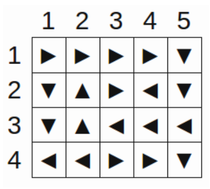
Sample Output

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

Problem: Magnetic Fields

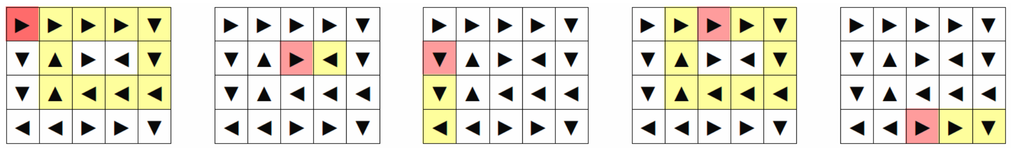


Bob has a new amazing board game based on a rectangular magnetic field which is divided into square cells. Below each cell there is a magnet forcing any metallic object to go towards another neighboring cell (to the north, south, east or west). The following image shows an example field with 4 rows and 5 columns:



	1	2	3	4	5
1	▶	▶	▶	▶	▼
2	▼	▲	▶	◀	▼
3	▼	▲	◀	◀	◀
4	◀	◀	▶	▶	▼

Bob drops a metallic ball on the board and watches the path it follows, forced by the magnets. Sometimes the ball falls off the board, while others times it stays on it until he gets tired of seeing it going round and round. The following image shows examples of initial drop cells (in red) and the path the ball follows afterwards (in yellow):



Bob is really interested in knowing how many different cells a ball traverses after being dropped on the board (including the initial cell). For example, in the first example of the figure above, the ball is dropped on cell (1,1) and traverses exactly 11 cells, before starting to repeat its path. On the other hand, in the example in the middle, a ball is dropped on cell (2,1) and it only traverses 3 different cells before it falls off the board.

Can you help Bob computing the lengths of these paths?

Task

Given a rectangular magnetic field with **R** rows and **C** columns, where each cells indicates a direction (north, south, east or west), and **Q** queries, each one indicating a starting position in the grid, your task is to compute, for each query, how many different cells would a ball traverse when following the directions indicated by the magnets.

Input

The first line contains two integers, **R** and **C**: the number of rows and columns of the magnetic field. **R** lines follow, each one with **C** characters indicating the direction associated with each cell: The characters are one of 'N' (north), 'S', (south), 'E' (east) or 'W' (west).

The following line contains an integer **Q**, indicating the number of queries. Each of the following **Q** lines has an individual query on the form of two integers Y_i and X_i, indicating that the starting point is on row Y_i and column X_i.

Constraints

$1 \leq \mathbf{R},\mathbf{C} \leq 300$	number of rows and columns
$1 \leq \mathbf{Q} \leq 25\,000$	number of queries
$1 \leq Y_i \leq \mathbf{R}$	starting row
$1 \leq X_i \leq \mathbf{C}$	starting column

Output

The output should have **Q** lines, each one indicating the number of cells the ball traverses when starting in the corresponding initial cell.

Sample Input

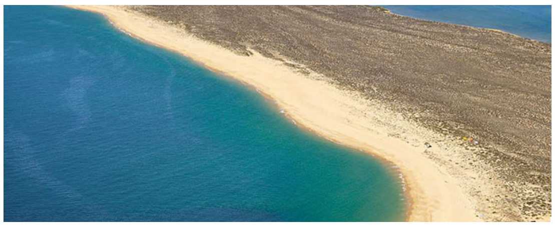
```
4 5
EEEES
SNEWS
SNWWW
WWEES
5
1 1
2 3
2 1
1 3
4 3
```

Sample Output

```
11
2
3
10
3
```

Note that the sample input corresponds to the figure in the problem statement.

Problem: Desert Beach



Not far from where I live, there is a desert beach. Not many people go there because it's rather difficult to reach: you either walk for two hours or risk your life crossing the lagoon in a decrepit, unsafe small boat and then walk half an hour. But it's a fantastic beach and a paradise for the few of us who are brave enough.

There's lots of space, and the people, or families, at the beach place their beach umbrellas far from each other. That's nice, because if you choose a desert beach it's also because you enjoy being on your own.

Still, at times, in the peak of summer, the number of people increases and I found it difficult to choose the best place for my beach umbrella in those days when I was not the first to arrive. This is because I want to honor the unwritten compromise that when the beach has more than a few people, each newcomer should stay as far as possible for any of the others who arrived before.

Since I am a programmer, I plan to write a program that, given the locations of the umbrellas of the people who have arrived before me, computes the location I should choose, in such a way that I am as far from others as possible.

I am still researching for my program. I modelled the beach as a rectangular grid. (Well, I am a programmer: I model things!) I decided that the first task is computing the set of locations in the grid which are furtherest from any location that has an umbrella in it, using Manhattan distance [1].

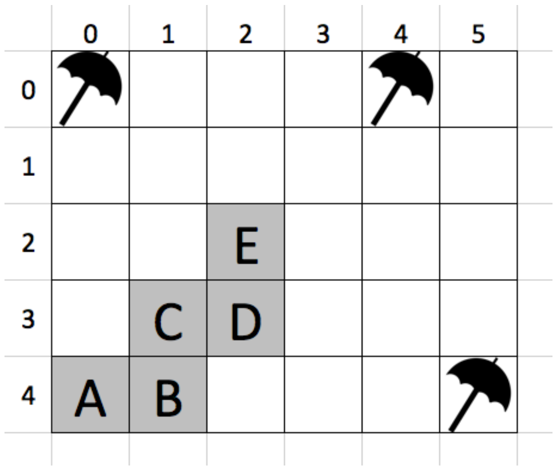
Then I could pick among these locations the one that suits me best. In a perfect world, the next person to arrive will do as I did, and everybody will be happy.

Say that upon arriving at the beach, I found a place that is D units removed from the nearest umbrella, D being the maximum possible, of course. (I assume there's one umbrella already, at least.) Perhaps the next person can find a place for herself also removed D units from the nearest umbrella (including mine); and perhaps she cannot. And perhaps she can or cannot depending on how I made my choice.

I am a good citizen, I do not want to curtail the choices of others. Therefore, if by choosing some of those locations furtherest from the nearest umbrella I decrease the maximum distance between any free grid cell and the nearest umbrella, and by choosing others, I do not, I definitely prefer the latter.

Here is an exemple, using a 6*5 beach with three umbrellas.

[1] Manhattan distance, you may recall, is the sum of the absolute values of the differences of the coordinates: $|x1 - x2| + |y1 - y2|$.



Any of the cells marked in grey are at a distance of 4 from the nearest umbrella. That's the best I can get. If I choose cell A or cell E, the next person will be able to find a spot 4 units away from the nearest umbrella (namely, E or A, respectively). But, if I choose B, C or D, there won't the any more free cells that are 4 units away from the nearest umbrella.

This observation means that what I really need is computing a partition of the set of locations furthest from any umbrella: the set of locations that, if chosen, do not decrease the maximum distance to the nearest umbrella, and the set of those that do.

Afterwards, if either set is empty, I will pick a location randomly from the set that is not empty; if both are non-empty, I will pick a location randomly from the set that does not decrease the maximum distance. But, that's trivial.

Task

Given a rectangular grid representing a beach, where some cells (at least one and not all) are occupied by beach umbrellas, partition the set of free cells that are furthest away from the nearest umbrella in two subsets: the subset of those cells which if they become occupied will not decrease the maximum distance from any cell to the nearest umbrella; and its complement.

Input

The first line of input contains two positive integers, W and H, representing the width and the height of the grid. The second line contains one non-negative integer, N, representing the number of grid cells that are occupied. N lines follow, each containing two non-negative integers, X and Y, representing the horizontal and vertical coordinates of each cell that is occupied.

Output

The first line of the output contains the number of cells in first subset of the partition; then, for each cell in that subset there is a line with the X and Y coordinates, separated by a space, ordered by Y and then by X. Then, the same thing for the other subset.

Constraints

Grid width, W	$W \leq 100$
Grid height, H	$H \leq 100$
Number of cells that are occupied, N	$0 < N < W \times H$
Horizontal coordinates, X	$0 \leq X < W$
Vertical coordinates, Y	$0 \leq Y < H$
Non duplicates	all pairs of coordinates in the input are unique.

Sample Input 1

This is the case used above.

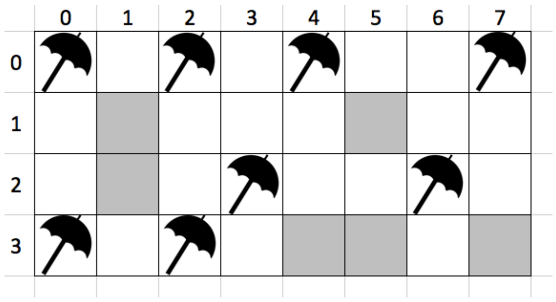
```
6 5
3
0 0
4 0
5 4
```

Sample Output 1

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

Sample Input 2

This case corresponds to the following illustration:



The distance from each of the grey cells to the nearest umbrella is 2. I can choose any of those, the next person will still be able to choose a cell 2 units away from the nearest umbrella.

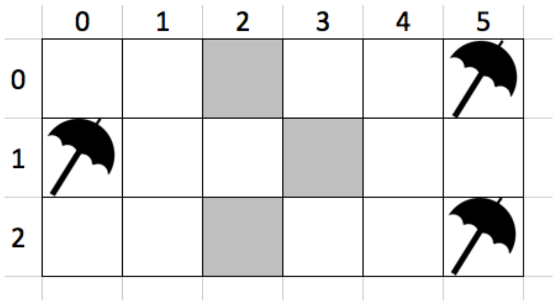
```
8 4
8
6 2
0 0
2 3
4 0
7 0
2 0
0 3
3 2
```

Sample Output 2

```
6
1 1
5 1
1 2
4 3
5 3
7 3
0
```

Sample Input 3

This case corresponds to the following illustration:



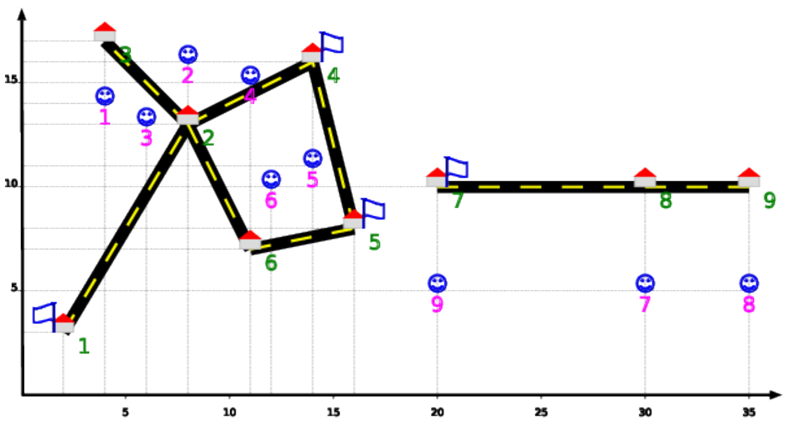
The distance from each of the grey cells to the nearest umbrella is 3. No matter which of these I choose, the next person will not be able to choose a cell 3 units away from the nearest umbrella.

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

Sample Output 2

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

Problem: Political Campaign 2



There is a new political campaign in Lagutrop. The leaders of the political parties have devised a new strategy to convince the **countryside** electorate to vote for them. They are going to travel around the country in their caravan talking to all people they encounter in their way.

Since the leaders have to follow their party line, leaders from left wing parties will seat on the left side of the van. They will talk to people that they see from the left window of the van, people that are straight ahead from the left window. Likewise, leaders from right wing parties will seat on the right side of the van and talk to people that are on the right side of the van. Moreover, all leaders have lack of vision, so they will miss out all people that are beyond their vision range.

All leaders **start** in their headquarter city and they take the first road that leaves their city (this is the **first road** in the **input section** that mentions the leader headquarter city). When they reach a new city, leaders from left wing parties will take the leftmost unvisited road (that makes the greatest angle from their current direction). Likewise, leaders from right wing parties will take the rightmost unvisited road. A left wing party will never take a road on his right. If upon reaching a city his only options are roads to his right, he ends his campaign. The corresponding stopping criteria applies to right wing leaders. If there are no roads available, he also ends his campaign.

Task

You should calculate how many people each leader talks to.

Input

The first line contains four integers: C, the number of cities; R, the number of roads; L, the number of leaders; E, the electorate size. Each of the following C lines contain two integers, (x_c, y_c) representing the coordinates of a city. Each of the following R lines contain two integers, c^r_1,c^r_2, representing two cities, the endpoints of road r. Each of the following L lines contain: either the character `L` for a leftwing party or the character `R` for a rightwing party; an integer representing the vision range, v; an integer, c^l representing the headquarter city of leader l. Each of the following E lines contain the coordinates of people, (x_e, y_e).

Constraints

$1 \leq C \leq 100$	number of cities
$1 \leq R \leq 100$	number of roads
$1 \leq L \leq 100$	number of leaders
$1 \leq E \leq 100$	electorate size
$1 \leq v \leq 3 \cdot 10^4$	leader vision range
$0 \leq x_c, y_c, x_e, y_e \leq 3 \cdot 10^4$	coordinates of cities and people
$1 \leq c^r_j, c^l \leq C$	road endpoints, headquarter city
$c^r_1 \neq c^r_2$	road endpoints are different

no two cities occupy the same position
departing roads from a city have unique directions
people are **not** in the cities **nor** on the roads
all arithmetic can be done with 64 bit integers
as with all heavy drugs, floating point arithmetic should be taken with moderation

Output

A single line with L integers separated by a single space, representing the number of people each one talks to.

Sample Input

This input example corresponds to the figure.

```
9 8 4 9
2 3
8 13
4 17
14 16
16 8
11 7
20 10
30 10
35 10
1 2
2 3
2 4
2 6
```

```
5 4
6 5
7 8
9 8
L 5 1
R 1000 5
R 2 4
R 5 7
4 14
8 16
6 13
11 15
14 11
12 10
30 5
35 5
20 5
```

Sample Output

```
2 0 1 3
```