

Hong Kong Olympiad in Informatics 2018/19 Senior Group

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
S191	Unstoppable Onslaught	1.000 s	256 MB	9 + 12 + 20 + 25 + 34
S192	Two Towers	1.000 s	256 MB	10 + 14 + 15 + 16 + 22 + 23
S193	Alice's Housekeeper	1.000 s	256 MB	8 + 21 + 39 + 32
S194	Snakes and Snakes	1.000 s	256 MB	3 + 4 + 13 + 24 + 19 + 37

Notice:

Unless otherwise specified, inputs and outputs shall follow the format below:

- One space between a number and another number or character in the same line.
- No space between characters in the same line.
- Each string shall be placed in its own separate line.
- Outputs will be automatically fixed as follows: Trailing spaces in each line will be removed and an end-of-line character will be added to the end of the output if not present. All other format errors will not be fixed.

C++ programmers should be aware that using C++ streams (`cin` / `cout`) may lead to I/O bottlenecks and substantially lower performance.

For some problems 64-bit integers may be required. In Pascal it is `int64`. In C/C++ it is `long long` and its token for `scanf` / `printf` is `%lld`.

All tasks are divided into subtasks. You need to pass all test cases in a subtask to get points.

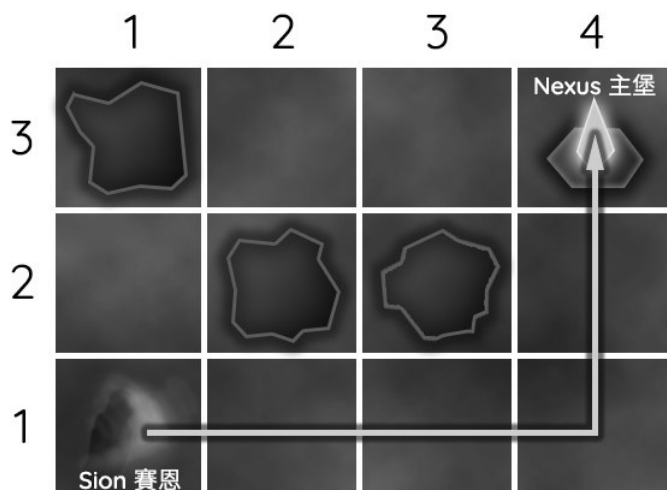
S191 - UNSTOPPABLE ONSLAUGHT

Time Limit: 1.000 s / Memory Limit: 256 MB

Riot Games recently created a new board game featuring two League of Legends champions Sion and Trundle. The board game requires 2 players and is played on a rectangular grid with R rows and C columns. The rows and columns are numbered from 1 to R from bottom to top and 1 to C from left to right respectively. We denote the cell on the row r , column c as cell (r, c) .

Your friend is playing as Sion, which is currently at cell $(1, 1)$. The goal of Sion is to cast his ultimate to reach your nexus at cell (R, C) . You, who are playing as Trundle, need to protect your nexus from Sion. The rules of the board game states that:

- $R \times C > 2$. In other words, Sion is currently not at your nexus or the cell just next to your nexus.
- There are originally N obstacles on the grid, which may help in protecting your nexus.
- Sion's ultimate has its weakness. The movement are restricted by the following rules: If at some moment it is at the cell (i, j) , it can only move:
 - upwards to the cell $(i + 1, j)$, only if this cell exists (i.e. $i + 1 \leq R$) and is not occupied by any obstacle; or
 - rightwards to the cell $(i, j + 1)$, only if this cell exists (i.e. $j + 1 \leq C$) and is not occupied by any obstacle.



You need to protect the nexus by placing zero or more obstacles at some of the empty cells, so that Sion cannot move to your nexus. At the same time, you should place as few obstacles as possible. Please note that each cell should contain no more than one obstacle (including those which are originally there and those you placed). Also, you cannot place any obstacle on the cell $(1, 1)$ nor (R, C) .

INPUT

The first line contains three integers, R , C and N .

Each of the next N lines consists of two integers x and y , stating that one of the obstacle is located on (x, y) . It is guaranteed that $1 \leq x \leq R$, $1 \leq y \leq C$, and the input cells are distinct. Also, $(x, y) \neq (1, 1)$ and $(x, y) \neq (R, C)$.

OUTPUT

The first line contains a single integer M , denoting the minimum number of obstacles you have to place so that Sion cannot move to your nexus.

For each of the M obstacles, output a line that consists of two integers x and y , stating that the obstacle will be placed on the cell located at row x , column y .

Any valid plan that places the minimum number of obstacles will be accepted.

It can be proven that under the given constraints, there always exist at least one solution.

SAMPLE TESTS

	Input	Output													
1	<table><tr><td>3</td><td>3</td><td>3</td></tr><tr><td>1</td><td>3</td><td></td></tr><tr><td>2</td><td>2</td><td></td></tr><tr><td>3</td><td>1</td><td></td></tr></table>	3	3	3	1	3		2	2		3	1		<table><tr><td>0</td></tr></table>	0
3	3	3													
1	3														
2	2														
3	1														
0															

Your nexus is protected without placing any additional obstacles as Sion cannot move to the cell (3,3).

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

This sample corresponds to the image in the problem statement.

It is easy to see that your nexus will be attacked by Sion if no additional obstacles are placed. By placing an obstacle at the cell (1,4), Sion can no longer move from (1,1) to (3,4).

Please note that (1,4) is not the only possible solution. Instead, you can place an obstacle at any of the three cells: (1,2), (1,3), (2,4), to protect your nexus.

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

SUBTASKS

For all cases:

$$1 \leq R, C \leq 2000$$

$$R \times C > 2$$

$$0 \leq N \leq \min(500000, R \times C - 2)$$

	Points	Constraints
1	9	$N = 0$
2	12	$R = C = 2$
3	20	$1 \leq R, C \leq 50$
4	25	$1 \leq R, C \leq 500$
5	34	No additional constraints



S192 - TWO TOWERS

Time Limit: 1.000 s / Memory Limit: 256 MB

Bitland is a country which has a structure of a $N \times M$ grid. The cell on the r -th row and the c -th column is labeled (r, c) , where $1 \leq r \leq N$, $1 \leq c \leq M$. There is exactly one household living at each cell.

After a devastating war with Byteland, most facilities in Bitland were destroyed and needed to be rebuilt. Rebuilding telecommunications tower is, of course, top-priority at the moment --- there is bound to be riot soon if residents cannot watch TV or use the phone!

Due to limits in budget and equipment, the best Bitland can do is to build two telecommunications towers A and B, where tower A has power P_A and tower B has power P_B . Each tower is to be built at one of the $N \times M$ cells (without needing any forced relocation of households). Note that the towers can be built at the same cell.

Suppose tower A is built at (r_A, c_A) and tower B at (r_B, c_B) . Each household can select exactly one tower to receive signal from. For the household located at (r, c) , if tower A is chosen, the signal strength is $\max(0, P_A - |r_A - r| - |c_A - c|)$. Similarly, if tower B is chosen, the signal strength is $\max(0, P_B - |r_B - r| - |c_B - c|)$. Being rational, each household will choose a tower to maximize signal strength.

Help Bitland government decide where to build the two towers, so as to maximize the minimum signal strength among all $N \times M$ households.

INPUT

The first and only line of input consists of four space-separated integers N , M , P_A and P_B .

OUTPUT

On the first line, output one single integer, the maximal value of the minimum signal strength among all $N \times M$ households.

On the second line, output four space-separated integers r_A, c_A, r_B, c_B with $1 \leq r_A, r_B \leq N$ and $1 \leq c_A, c_B \leq M$, so that the maximal value can be attained by placing tower A at (r_A, c_A) and tower B at (r_B, c_B) .

If there are multiple answers, output any of them.

SAMPLE TESTS

	Input	Output
1	3 5 7 0	4 2 3 1 1

If tower A is placed at (2, 3) and tower B at (1, 1), the signal strength is minimal at cells (1, 1), (1, 5), (3, 1), and (3, 5), with a value of 4. It is impossible to attain a minimal signal strength of 5 or higher.

In fact, tower B can be placed anywhere (including cell (2, 3)) without affecting the answer.

2	5 3 10 10	8 4 2 2 2
---	-----------	--------------

3	2 13 10 7	5 1 5 2 12
---	-----------	---------------

4	1 2 6 6	6 1 1 1 2
---	---------	--------------

5	4 2 1 3	1 1 2 3 1
---	---------	--------------

SUBTASKS

For all cases:

$$1 \leq N, M \leq 5 \times 10^8$$

$$0 \leq P_A, P_B \leq 10^9$$

	Points	Constraints
1	10	$P_A = 0$
2	14	$N = 1$
3	15	$N = 2$ $M \geq 2$
4	16	$1 \leq N, M \leq 10$
5	22	$1 \leq N, M \leq 2000$
6	23	No additional constraints

S193 - ALICE'S HOUSEKEEPER

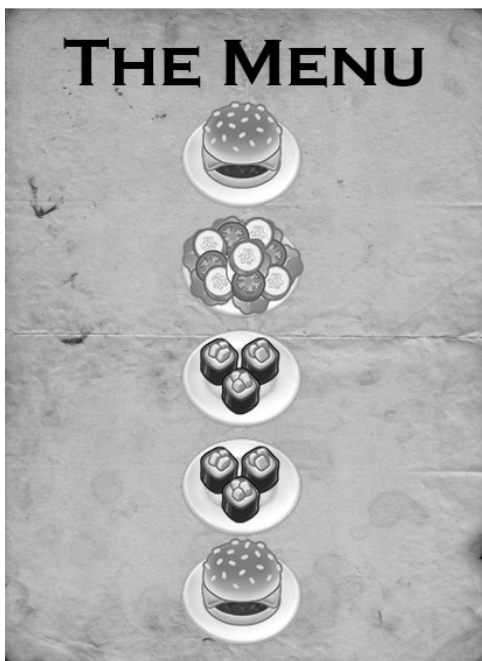
Time Limit: 1.000 s / Memory Limit: 256 MB

Because of the coming exam period, Alice feels very stressful and gets mad easily. When she is angry, she will scold at her housekeepers, even worse she may punish them!













You, as one of the housekeepers of Alice, do not want to get punished. One way to keep your master in a good mood is to serve her favorite dishes as dinner. You have got a list of K dishes Alice likes, denoted as the first K lowercase letters (a, b, c...) To save time thinking what to serve, you have figured out a great idea.

You order the chef to write a cyclic menu with N dishes, each of which must be one of the K dishes Alice likes. On the first day, you will serve Alice the first dish in the menu. On each of the following days, you will serve Alice the succeeding dish. After serving the N^{th} dish, the process will be repeated by serving Alice the first dish on the upcoming day.

For example, if there is a cyclic menu with 5 dishes abcca:



On the first 12 days, Alice will be served with the dishes abccaabccaab:

1	2	3	4	5	6	7	8	9	10	11	12
											

(Dishes from the game Overcooked!)

Hurray! You think you will be successful in pleasing your master. Maybe... Alice will reward you some nice presents for Christmas!

Two hours later, you suddenly recall the painful experience in the past. You served the same spaghetti for two consecutive days and got seriously punished...

7 p.m. in the dining room...

Alice: Spaghetti!? Again? Rich people like me never eat the same food for two consecutive days!!!

You: *Internally screaming*



You have completely forgotten that Alice hates having the same dish as yesterday! Doing something she does not like definitely angers her! Unfortunately, the chefs may not know this fact and may have arranged the menu with adjacent and identical dishes. You know a nice menu should guarantee that no same dish would be served in any two consecutive days, so the menu made by the chefs may be unsatisfactory.

You are lazy, so you want to adopt the chefs' menu with the least efforts in amending it. To satisfy Alice's strange requirements, you are going to replace some of the dishes by any of the dishes Alice likes. Given the chefs' menu, how many replacements do you need to make on the menu at least to not anger Alice? If there is no way you can amend the menu to fulfill the requirements, output `Impossible`.

INPUT

The first line contains two integers, N, K .

The second line contains N characters. The i^{th} character represents the i^{th} dish in chefs' menu.

OUTPUT

If it is impossible to please Alice with the menu even after modifications, output `Impossible`.

If it is possible to do so, output two lines. On the first line, output a single integer in a line denoting the minimum number of replacements needed.

On the second line, output N characters showing the new menu you have made.

If there are more than one menu that satisfies the given requirements, output any of them.

SAMPLE TESTS

	Input	Output
1	8 26 abzzabzz	2 abyzabzy
2	6 2 aaabbb	2 ababab
3	3 2 bbb	Impossible
4	5 26 qqweq	1 rqweq

There are only 8 possible menus with length 3: `aaa`, `aab`, `aba`, `abb`, `baa`, `bab`, `bba`, `bbb`.
All of them will cause repeated dish in some consecutive two days.

`fqweq` and `dqweq` are some of the other possible answers with minimum number of replacement.

Please note that `qrweq` is incorrect as it will cause Alice having the same dish on, for example, day 5 and day 6.



SUBTASKS

For all cases:

$$1 \leq N \leq 10^6$$

$$2 \leq K \leq 26$$

	Points	Constraints
1	8	$N = 2$
2	21	$K = 2$
3	39	$1 \leq N \leq 5000$
4	32	No additional constraints



S194 - SNAKES AND SNAKES

Time Limit: 1.000 s / Memory Limit: 256 MB

On a cold Christmas Eve, all of your friends are dating with their girlfriends and no one wants to date with you, so you have no choice but to stay at home and play “Snakes and Snakes” alone :(

The board has $N + 1$ squares, numbered from 0 to N . Your piece starts at square 0 and needs to reach square N .

There are a total of M snakes on the board. Snakes bring your piece from an “entry square” with a higher number to an “exit square” with a lower number.

You play with a dice with numbers from 1 to K . In a round, you “row” the dice and move your piece forward by d squares, where d is the number displayed on the dice. If the square your piece moves to is the entry square of a snake, it moves to the exit square of that snake.

It is guaranteed that the entry square of a snake will not coincide with the exit square of another snake.

Since there is no competitor, you decide to just finish the game in as few dice rows as possible. Also, since you have no friends to play with you, you can cheat and choose any possible numbers on the dice. What is the minimum number of moves to finish the game? (It is possible that there is no way to finish the game! What a lonely Christmas...)

INPUT

The first line consists of three integers N , M and K .

Each of the next M lines consists of two integers x and y , stating that there is a snake from square x to square y . It is guaranteed that:

- $0 \leq y < x < N$
- the entry square of a snake will not coincide with the exit square of another snake
- the entry squares are distinct

The snakes are given in ascending order of their entry square.

OUTPUT

If there is no way to finish the game, output Lonely Christmas.

Otherwise, output the minimum number of moves needed to reach square N .

SAMPLE TESTS

	Input	Output										
1	<table><tr><td>7</td><td>0</td><td>3</td></tr></table>	7	0	3	<table><tr><td>3</td></tr></table>	3						
7	0	3										
3												
2	<table><tr><td>3</td><td>1</td><td>1</td></tr><tr><td>2</td><td>1</td><td></td></tr></table>	3	1	1	2	1		<table><tr><td>Lonely Christmas</td></tr></table>	Lonely Christmas			
3	1	1										
2	1											
Lonely Christmas												
3	<table><tr><td>6</td><td>2</td><td>2</td></tr><tr><td>2</td><td>0</td><td></td></tr><tr><td>4</td><td>0</td><td></td></tr></table>	6	2	2	2	0		4	0		<table><tr><td>4</td></tr></table>	4
6	2	2										
2	0											
4	0											
4												

The fastest way is $0 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 6$.

SUBTASKS

For all cases:

$$1 \leq K \leq N \leq 10^9$$

$$0 \leq M \leq \min(N - 1, 200000)$$

	Points	Constraints
1	3	$M = 0$
2	4	$K = 1$
3	13	$K = 2$ $2 \leq N \leq 200000$
4	24	$K = 2$
5	19	$K \leq 100$ $N \leq 200000$
6	37	No additional constraints