Hong Kong Olympiad in Informatics 2024/25 Senior Group

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
S251	Path of Fortune	1.000 s	256 MB	7 + 9 + 13 + 27 + 20 + 24
S252	Digit Puzzle	2.000 s	256 MB	4 + 4 + 8 + 25 + 18 + 15 + 26
S253	Miles Away	1.500 s	256 MB	5 + 10 + 13 + 23 + 6 + 30 + 13
S254	Faultline of the Earthquake	1.000 s	256 MB	7 + 11 + 9 + 40 + 29 + 4

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 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.

S251 - PATH OF FORTUNE

Time Limit: 1.000 s / Memory Limit: 256 MB

Alice loves playing games! One day, she found a new game named "Path of Fortune", and she started playing it.

At the start of the game, the player will be given R red cards, B blue cards and an infinite amount of tokens. In the game, the player is presented with Q *decisions* to make, and each decision has two *options* of same or different types available for the player to choose from. When an option is chosen, the player has to perform an action specified by the option. The following are the three different types of options in the game:

- Type `R`: If this option is picked, the player has to discard a red card.
- Type `B`: If this option is picked, the player has to discard a blue card.
- Type `T`: An additional parameter x will also be given for each option of this type. If this option is picked, the player has to pay x tokens.

If the player runs out of cards of a colour, the corresponding option type will then be deemed invalid and cannot be chosen for the rest of the game. If both options in a decision are invalid, the player loses the game immediately. To prevent the game devolving into a game of luck, the players will be told the options in all Q decisions at the start so they can strategize beforehand.

Being a competitive programmer, Alice cannot help but wonder: given the deck of cards she will receive, as well as the options in all decisions, can she determine if she could win, and if so, what is the least amount of tokens she will have to pay? Please help Alice, who is too busy playing the game, figure out the answer!

INPUT

The first line contains two integers R and B, the number of red and blue cards Alice initially gets.

The second line contains an integer Q, the number of decisions Alice has to make.

The i-th line of the next Q lines contains the two strings, the two options presented in the i-th decision. Options are presented in the following format:

- `R`: an option of type `R`
- `B`: an option of type `B`
- `T-x`: an option of type `T` with a parameter x

OUTPUT

If Alice cannot win the game no matter how she picks the options, output `-1`.

Otherwise, output the minimum number of tokens Alices needs to pay in order to win.

SAMPLE TESTS

1

Input	Output
1 3 4 R T-10 B R B R T-100 R	10

Alice can pick the options in the following way:

- Choose the second option for decision 1 and pay 10 tokens.
- Choose the first option for decision 2 and discard a blue card. Alice has 2 blue cards left.
- Choose the first option for decision 3 and discard a blue card. Alice has 1 blue card left.
- Choose the second option for decision 4 and discard a red card. Alice has no red cards left.

By choosing this way she can win the game while paying 10 tokens. It can be proven that this is the minimum amount of tokens she has to pay to win the game.

2

Input	Output
1 3 4 R T-10 R R B R T-100 R	110

Alice can pick the options in the following way:

- Choose the second option for decision 1 and pay 10 tokens.
- Choose the second option for decision 2 and discard a red card.
  Alice has no red cards left, hence she cannot choose type `R` options in subsequent decisions.
- As Alice has no red cards left, she has to choose the first option for decision 3 and discard a blue card. Alice has 2 blue cards left.
- As Alice has no red cards left, she has to choose the first option for decision 4 and pay 100 tokens.

By choosing this way she can win the game while paying 110 tokens. It can be proven that this is the minimum amount of tokens she has to pay to win the game.

3

Input	Output
1 1 4 R R B B B R R B	-1

SUBTASKS

For all cases:
$0 \le R, B \le 10^5$
$1 \le Q \le 10^5$
All options are either type T, type R or type B
$0 \le x \le 10^4$ for all type T options

	Points	Constraints
1	7	$Q = 1$ All options are type T
2	9	$Q = 1$ All options are either type T or type R
3	13	All options are type T
4	27	All options are either type R or type B
5	20	All options are either type T or type R
6	24	No additional constraints

S252 - DIGIT PUZZLE

Time Limit: 2.000 s / Memory Limit: 256 MB

Bob likes solving puzzles, especially puzzles with numbers. Today, he is presented with a *digit puzzle* which consists of $N + 1$ blank cells separated by N signs, shown as follows:

	*		+		*		-		*		=	

The N signs are denoted as $S_1, S_2, \dots, S_N$ from left to right. $S_1, S_2, \dots, S_{N-1}$ only consists of the operators `+` (addition), `-` (subtraction) and `*` (multiplication), and S_N is always `=` (the equality sign). Bob's goal is to fill in a single digit into each blank cell such that the equality holds.

"This is stupid!" Bob says. "We can just fill the digit `0` into each blank cell!"

0	*	0	+	0	*	0	-	0	*	0	=	0

To give himself some extra challenge, Bob wonders if it is possible to find a solution to the puzzle **using the digits from `1` to `9` (inclusive) only** (i.e. without using the digit `0`).

For example, the following shows a solution to the puzzle above since $3 \times 5 + 1 \times 7 - 2 \times 9 = 15 + 7 - 18 = 4$:

3	*	5	+	1	*	7	-	2	*	9	=	4

Write a program to solve the puzzles for him.

INPUT

The first line of input contains a single integer N, the number of signs in the puzzle.

The second line of input contains a string S of N characters, where $S_1, S_2, \dots, S_N$ are the signs in the puzzle from left to right.

It is guaranteed that the result of the expression on the left of the equality sign is always between -10^9 to 10^9 (inclusive), regardless of the digits (from `0` to `9`) filled into the puzzle.

OUTPUT

If there are no solutions to the puzzle which uses the digits `1` to `9` only, output `Impossible`.

Otherwise, output `Possible` on the first line. Output $N + 1$ single-digit integers from `1` to `9` on the second line, a solution to the puzzle. You should output the filled-in digits in the puzzle from left to right.

If there are multiple solutions, you may output any of them.

SAMPLE TESTS

	Input	Output
1	6 `*+*-*=`	Possible 3 5 1 7 2 9 4

This sample test matches the example shown in the problem statement.

	Input	Output
2	6 `*+*-*=`	Possible 2 3 1 2 2 3 2
3	12 `+++++++++++=`	Impossible

SUBTASKS

For all cases:
$2 \leq N \leq 10^6$
$S_1, S_2, \ldots, S_{N-1}$ only consists of `+`, `-` or `*`
$S_N =$ `=`
Regardless of the digits (from `0` to `9`) filled into the puzzle, the result of the expression on the left of the equality sign is always between -10^9 and 10^9 (inclusive).

	Points	Constraints
1	4	S only consists of `+` and `=`
2	4	S only consists of `-` and `=`
3	8	S only consists of `+`, `*` and `=`
4	25	S only consists of `+`, `-` and `=`
5	18	$S_1 =$ `*` $S_2 = S_3 = \ldots = S_{N-1} =$ `-`
6	15	$N \geq 3$ There exists some $1 \leq k < N - 1$ such that: $S_1 = S_2 = \ldots = S_k =$ `*` $S_{k+1} = S_{k+2} = \ldots = S_{N-1} =$ `-`
7	26	No additional constraints

S253 - MILES AWAY

Time Limit: 1.500 s / Memory Limit: 256 MB

It has been three months since you left Byteland to continue your high school studies elsewhere. On a cold December day, all of your classmates are revising for their mid-year exams with their best friends and no one wants to revise with you, so you decide to call your best friend Alice back in faraway Byteland to organise a virtual study session. However, the time zone difference between the two of you poses a challenge - will you still manage to make it work?

In this world, a day is divided into N time blocks, numbered from block 0 to block $N-1$. The timeline is **cyclic**, meaning block 0 of the next day comes after block $N-1$ of the current day. The time zone you are in is **slower** than Byteland, where Alice is in, by d blocks (where $0 \le d < N$). Formally, if the current time **in your local timezone** is block x (where $0 \le x < N$), the corresponding time **in Byteland** is block $(x+d) \bmod N$.

Alice's daily schedule, as well as your daily schedule, are given as follows:

- Alice has M_A daily events, during which she is unable to attend the study session. The i-th event ($1 \le i \le M_A$) spans from block $L_{1,i}$ to block $R_{1,i}$ (inclusive) **in Byteland time**.
- You have M_B daily events, during which you are unable to attend the study session. The i-th event ($1 \le i \le M_B$) spans from block $L_{2,i}$ to block $R_{2,i}$ (inclusive) **in your time zone**. Transformed **into Byteland time**, the event spans from block $(L_{2,i}+d) \bmod N$ to block $(R_{2,i}+d) \bmod N$.

Each event is at most N blocks (1 day) long. For example, when $N=10$, an event spanning blocks 4 to 6 is 3 blocks long, not 13, 23, ...

Events may wrap around midnight. For example, when $N=10$, an event spanning blocks 8 to 1 includes blocks 8, 9, 0, 1.

You wish to organise a virtual study session with Alice that lasts for K **continuous time blocks**, **possibly wrapping around midnight**, ensuring both you and Alice are **free for the entire period**. Since you are rather desperate, you are willing to cancel **at most T of your scheduled events** to make time for the study session. However, Alice's events are **fixed** and **cannot be altered.**

You are given Q hypothetical queries, where the i-th query specifies an integer D_i. You must answer: assuming your time zone is **slower** than Byteland by $d=D_i$ blocks, is it possible for you to cancel **at most T of your scheduled events** to schedule a meeting of K consecutive time blocks where both of you are free?

And so, though miles away, you strive to find a shared moment in time, where schedules align and studying together eases your loneliness.

INPUT

The first line of the input contains two integers N and K.

The next line of the input contains two integers M_A and M_B.

The next M_A lines contain descriptions of Alice's events. The i-th ($1 \le i \le M_A$) of these lines comes in the form of two integers, $L_{1,i}$ and $R_{1,i}$, describing the range for the i-th event. All M_A of Alice's events do not overlap with each other.

The next M_B lines contain descriptions of your events. The i-th ($1 \le i \le M_B$) of these lines comes in the form of two integers, $L_{2,i}$ and $R_{2,i}$, describing the range for the i-th event. All M_B of your events do not overlap with each other.

The next line of the input contains a single integer T.

The next line of the input contains a single integer Q.

The next line contains Q integers $D_1, D_2, \ldots, D_Q$.

OUTPUT

Output a string of length Q. The i-th character ($1 \leq i \leq Q$) should be `1` if the answer to the i-th query is Yes, or `0` otherwise.

SAMPLE TESTS

	Input	Output
1	12 3 2 2 9 11 2 4 2 4 0 1 1 3 0 4 5	110

In this sample, you want to schedule a study session that lasts for $K = 3$ continuous time blocks. You can cancel **at most** $T = 1$ event.

In the first query, $D_1 = 0$. In Byteland time, your events occur at blocks 2, 3, 4 and 0, 1 respectively. You may hold the meeting from block 6 to block 8 (in Byteland time). No events are cancelled, so the answer to the query is Yes.

In the second query, $D_2 = 4$. In Byteland time, your events now occur at blocks 6, 7, 8 and 4, 5 respectively. If you cancel the event occupying blocks 6, 7, 8, you may hold the meeting from block 6 to block 8 (in Byteland time). Only 1 event is cancelled, so the answer to the query is Yes.

In the third query, $D_3 = 5$. In Byteland time, your events now occur at blocks 7, 8, 9 and 5, 6 respectively. It is impossible to schedule the study session with at most 1 event cancelled. Therefore, the answer to the query is No.

	Input	Output
2	24 15 1 0 13 21 0 5 5 1 2 23 15	11111

	Input	Output
3	10 6 1 1 8 1 5 4 1 1 0	1

Note that events may wrap around midnight.

SUBTASKS

For all cases:
$1 \le K \le N \le 10^9$
$0 \le M_A, M_B \le 700$
$0 \le L_{1,i}, R_{1,i} < N$ for $1 \le i \le M_A$
$0 \le L_{2,i}, R_{2,i} < N$ for $1 \le i \le M_B$
All M_A of Alice's events do not overlap with each other
All M_B of your events do not overlap with each other
$0 \le T \le M_B$
$1 \le Q \le 5 \times 10^5$
$0 \le D_i < N$ for $1 \le i \le Q$

	Points	Constraints
1	5	$T = 0$ $N \le 50$ $M_A = 1$ $M_B = 0$ $L_{1,1} \le R_{1,1}$ $Q \le 50$
2	10	$T = 0$ $N \le 50$ $M_A, M_B \le 50$ $Q \le 50$
3	13	$T = 0$ $M_B = 0$
4	23	$T = 0$ $Q \le 2000$ $K = 1$
5	6	$T = 0$ $Q \le 2000$
6	30	$T = 0$
7	13	No additional constraints

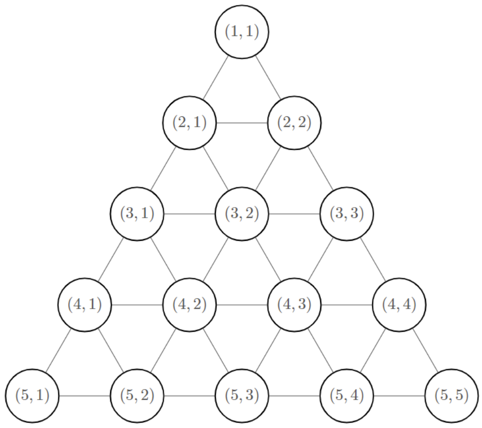
S254 - FAULTLINE OF THE EARTHQUAKE

Time Limit: 1.000 s / Memory Limit: 256 MB

Earthquakes are frequent in Byteland, due to its special ("weird") geographical structure.

The structure of Byteland is analogous to a triangular grid with N rows. The rows are numbered from 1 to N from top to bottom, so the row at the top is row 1, and the row at the bottom is row N. For all $1 \le i \le N$, on row i, there are exactly i cities, numbered from 1 to i from left to right. We denote the city c on row r as city (r, c).

Some pairs of cities in Byteland are connected by bidirectional roads. For each city (r, c) that is not located on the last row ($1 \le c \le r < N$), it is connected to two cities $(r + 1, c)$ and $(r + 1, c + 1)$ on the next row. Besides, for each city (r, c) that is not the last city of the row ($1 \le c < r \le N$), it is connected to the city $(r, c + 1)$. The figure on the right shows the map of Byteland when $N = 5$.



(The map of Byteland when $N = 5$)

We define the *distance* from a city (r_1, c_1) to another city (r_2, c_2) as the minimum number of roads that has to be travelled to reach (r_2, c_2) from (r_1, c_1). For example, when $N = 5$, the distance from city $(4, 1)$ to city $(2, 2)$ is 3, as one can travel via the path $(4, 1) \to (4, 2) \to (3, 2) \to (2, 2)$.

Now, an earthquake has happened! The earthquake happens along a *faultline* which is a path $(R_1, C_1) \to (R_2, C_2) \to \cdots \to (R_K, C_K)$, where $1 \le K \le N$. R, C and K are unknown to you. For all $1 \le i < K$, there must be a road connecting the cities (R_i, C_i) and (R_{i+1}, C_{i+1}). Faultlines are not unnecessarily bent, which means that the faultline follows a shortest path from (R_1, C_1) to (R_K, C_K). Formally, the distance between (R_1, C_1) and (R_K, C_K) must be $K - 1$.

While you have no information on the faultline of the earthquake, you have installed M sensors in M different cities. The i-th sensor is installed in the city (X_i, Y_i), which measures the *damage* of a city due to this earthquake. The damage of a city is defined as the shortest distance from the city to any city lying on the faultline of the earthquake. It is known that the reading of the i-th sensor is D_i.

Given the locations and the readings of the M sensors, you are now asked to recover a possible faultline of the earthquake, which is consistent with all the given information. It is guaranteed that a solution exists.

INPUT

The first line of input consists of two integers N and M, the number of rows in the triangle grid in the map of Byteland, and the number of sensors installed.

The next M lines each consists of three integers X_i, Y_i and D_i, denoting that the damage at city (X_i, Y_i) is D_i. There is at most one sensor at each city.

OUTPUT

Output an integer K on the first line, where $1 \le K \le N$, the number of cities on the faultline of the earthquake.

Next, output K more lines. The i-th line should contain the integers R_i and C_i, the i-th city in the faultline.

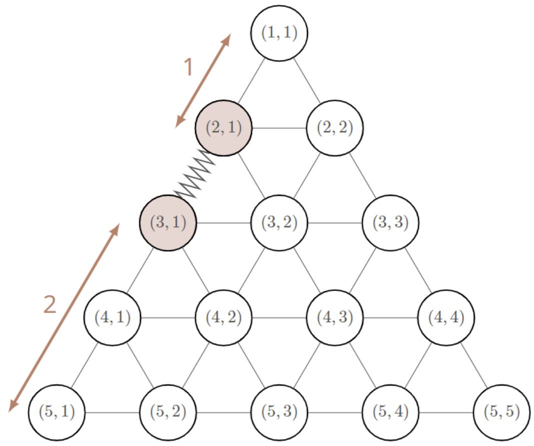
If there are multiple solutions with different K, R or C, output any of them.

SAMPLE TESTS

1

Input	Output
5 2 5 1 2 1 1 1	2 3 1 2 1

The figure below shows the faultline in the sample output:

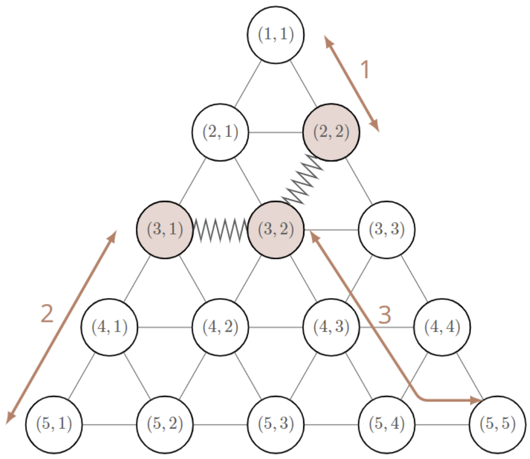


Starting from city $(5, 1)$, city $(3, 1)$ is the nearest city on the faultline. As the shortest distance between the cities is 2, the damage at city $(5, 1)$ is 2.
Starting from city $(1, 1)$, city $(2, 1)$ is the nearest city on faultline. As the shortest distance between the cities is 1, the damage at city $(1, 1)$ is 1.
Note that there exists other acceptable solutions.

2

Input	Output
5 3 1 1 1 5 1 2 5 5 3	3 3 1 3 2 2 2

The figure below shows the faultline in the sample output:



Starting from city $(1, 1)$, city $(2, 2)$ is the nearest city on the faultline. The damage at city $(1, 1)$ is 1.
Starting from city $(5, 1)$, city $(3, 1)$ is the nearest city on the faultline. The damage at city $(5, 1)$ is 2.
Starting from city $(5, 5)$, city $(3, 2)$ is the nearest city on the faultline. The damage at city $(5, 5)$ is 3.
However, the faultline $(3, 1) \to (2, 1) \to (3, 2)$ is invalid. The distance between cities $(3, 1)$ and $(3, 2)$ is 1, while the length of faultline is 2.

SUBTASKS

For all cases:

$3 \leq N \leq 10^5$

$1 \leq M \leq 10^5$

$1 \leq Y_i \leq X_i \leq N$ for $1 \leq i \leq M$

$0 \leq D_i < N$ for $1 \leq i \leq M$

$(X_i, Y_i) \neq (X_j, Y_j)$ for $1 \leq i < j \leq M$

	Points	Constraints
1	7	$N \leq 100$ $M \geq 2$ $D_i = 0$ for $1 \leq i \leq M$ $X_1 = 1$ $X_2 = N$
2	11	$N \leq 100$ $M \geq 2$ $D_i = 0$ for $1 \leq i \leq M$
3	9	$N \leq 100$ $M = 1$
4	40	$M \geq 2$ $X_1 = 1$, $D_1 = 0$ $X_2 = N$, $D_2 = 0$
5	29	$D_i = 1$ for $1 \leq i \leq M$ $1 < Y_i < X_i < N$ for $1 \leq i \leq M$
6	4	No additional constraints