Maratona de Programação da SBC 2020

This problem set is used in simultaneous contests:
Maratona de Programação da SBC 2020
Primera Fecha Gran Premio de México 2020
Tercera Fecha Gran Premio de Centroamérica 2020
Torneo Argentino de Programación 2020

November 14th, 2020

Problems book

General Information

This problem set contains 15 problems; pages are numbered from 1 to 25, without considering this page. Please, verify your book is complete.

A) Program name
1) Solutions written in C/C++ and Python, the filename of the source code is not significant, can be any name.
2) Solutions written in Java, filename should be: *problem_code*`.java` where *problem_code* is the uppercase letter that identifies the problem. Remember in Java the main class name and the filename must be the same.
3) Solutions written in Kotlin, filename should be: *problem_code*`.kt` where *problem_code* is the uppercase letter that identifies the problem. Remember in Kotlin the main class name and the filename must be the same.

B) Input
1) The input must be read from *standard input*.
2) The input is described using a number of lines that depends on the problem. No extra data appear in the input.
3) When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input.
4) Every line, including the last one, ends with an end-of-line mark.
5) The end of the input matches the end of file.

C) Output
1) The output must be written to *standard output*.
2) When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output.
3) Every line, including the last one, must end with an end-of-line.

Promo:

Sociedade Brasileira de Computação

v1.0

Problem A
Sticker Album

The Sticker Album of the ICPC 2020 Nlogonian Subregional just came out! Competitive programming hooligans all over the country are buying albums and collecting stickers, to celebrate the competition.

This sticker album is special because all stickers are equal: a picture of this year's trophy. To complete the album, you just need to collect enough stickers to fill all the slots in it.

You may be asking yourself: where is the fun in collecting those stickers? Well, to make things interesting, the stickers are sold in packets, each with a random number of stickers! Fans celebrate when they find a high number of stickers in a packet, make fun of those who got unlucky and found low numbers of stickers, and brag about filling their whole albums with just a few packets.

You just acquired your own album, and want to start filling it! But before buying your first sticker packets, you wondered: on average, how many packets does one need to buy in order to fill an album?

Input

The only input line contains three integers N, A and B, separated by a single space, satisfying $1 \leq N \leq 10^6$, $0 \leq A \leq B \leq 10^6$ and $B > 0$, where:

- N is the number of stickers it takes to fill an album;
- A is the minimum number of stickers in a packet;
- B is the maximum number of stickers in a packet.

The number of stickers in each packet is uniformly distributed in the closed interval $[A, B]$.

Output

The output consists of a single line, which must contain the expected number of packets it takes to complete an album. The number will be considered correct if it is within an absolute or relative error of 10^{-5} of the correct answer.

Input example 1	**Output example 1**
`40 0 2`	`40.33333`

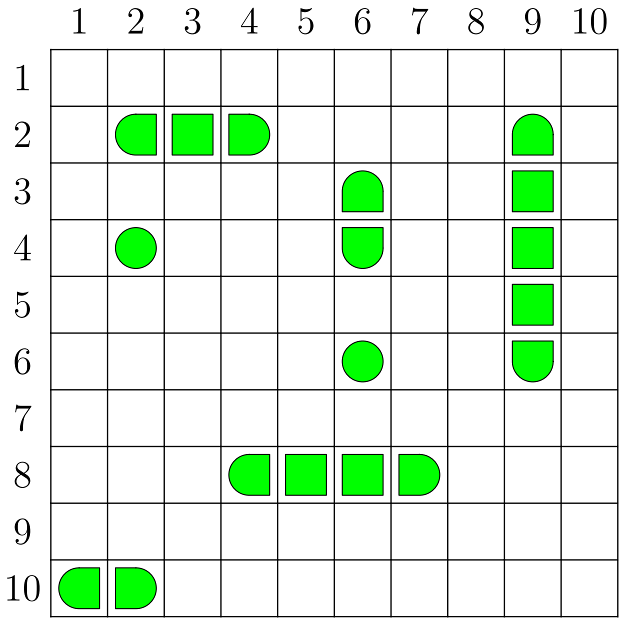
Input example 2	**Output example 2**
`100 1 10`	`18.72727`

Input example 3	**Output example 3**
`30 3 3`	`10.00000`

Input example 4	**Output example 4**
`314 5 8`	`48.74556`

Problem B
Battleship

Battleship is a classic strategy game for two players. Each player places his set of ships in a 10 × 10 grid and then games consist of guessing positions of ships. The actual game rules are not important for this problem, and many variations exist, but here we are interested on a much more basic problem. Given the lists of ships and their positions, your task is to check whether the initial positioning is valid.



The grid rows and columns are numbered from 1 to 10, and ships are positioned horizontally or vertically, occupying a contiguous sequence of squares of the board. For this problem, a positioning is valid if

- No position is occupied by more than one ship;
- All the ships are totally contained inside the board.

Input

The first line of the input contains an integer N, $1 \leq N \leq 100$ the number of ships. Each of the next N lines contain four integers D, L, R and C with $D \in \{0, 1\}$, $1 \leq L \leq 5$ and $1 \leq R, C \leq 10$ describing a ship. If $D = 0$ then the ship is aligned horizontally, and occupy positions $(R, C) \ldots (R, C + L - 1)$. Otherwise the ship is aligned vertically, occupying positions $(R, C) \ldots (R + L - 1, C)$.

Output

Output a single line containing a single character. If the initial positioning of the ships is valid, then write the upper case letter '`Y`'; otherwise write the uppercase letter '`N`'.

Input example 1	**Output example 1**

3
0 5 1 1
1 5 2 2
0 1 3 3
	Y

Input example 2	**Output example 2**

2
0 2 1 1
1 1 1 2
	N

Input example 3	**Output example 3**
1 0 2 10 10	N

Input example 4	**Output example 4**

7
0 3 2 2
1 5 2 9
1 2 3 6
1 1 4 2
0 1 6 6
0 4 8 4
0 2 10 1
	Y

Problem C
Concatenating Teams

Pepito is an ICPC coach who very often gets to "concat" the names of two existing teams, such as "AJI" and "Oxidados", in order to produce names for new teams, such as "AJIOxidados".

Since Pepito coaches teams from two different universities where he teaches, he had an idea: he will consider all possible such concatenations of a team name from university A, with a team name from university B (always in that order: first the one from university A, then the one from university B).

So, if team names from university A are "Buen" and "Kilo", and team names from university B are "Pan" and "Flauta", all the possible concatenations that he considers are the strings "BuenPan", "BuenFlauta", "KiloPan" and "KiloFlauta".

Furthermore, he calls a certain team *peculiar* if removal of that team makes the set of concatenations lose all the concatenations that used the name of that team.

In effect, in the previous example all the teams are peculiar. If however we consider names from A to be "xx" and "xxy" and names from B to be "z", "yz" and "xx", then "xx" from university A is not peculiar, because the name "xx" + "yz" = "xxyz" = "xxy" + "z" and can thus be formed without the need to use "xx" from A. For the same reason, "yz", "xxy" and "z" are not peculiar names. The only peculiar name in this case is "xx" from university B, because it is involved in creating the names "xxxx" and "xxyxx", and it is completely impossible to create any of these without using "xx" from university B.

Given the team names from both universities, your task is to calculate how many peculiar teams are there in each university.

Input

The first line contains two integers, M and N, separated by a space. The number of teams from university A is M and the number of teams from university B is N.

The second line contains the names of the teams from university A, separated by a space. The third line contains the names of the teams from university B, separated by a space.

All team names consist solely of lower case letters of the English alphabet.

No two teams from the same university have the same name.

$1 \leq M, N \leq 10^5$ and the total length of all team names is at most 10^6.

Output

Output a line containing two integers: the number of peculiar teams from university A, and the number of peculiar teams from university B, separated by one space.

Input example 1	**Output example 1**
2 2 buen kilo pan flauta	2 2

Input example 2	**Output example 2**
2 3 xx xxy z yz xx	0 1

Problem D
Divisibility Dance

In the country of Nlogonia, the citizens perform a special dance to worship the God of divisibility. The dance is performed by N men and N women arranged in two concentric circles. The men position themselves on the inner circle, and the women on the outer circle. Every women starts out in front of some man.

The dance consists of K steps; the first step is taken by the men, the second by the women and so on. At the i-th step, the people whose turn it is rotate P_i positions in the clockwise direction, while the people in the other circle stay put. In this way, each person changes partner to someone P_i positions away. A step is valid if every person's partner is different before and after the step AND if no pair of people ever face each other at two different moments in the process.

As part of the ritual, the dance must finish by forming pairs whose sum of ages leave the same remainder when divided by the sacred number M. In other words, if the sum of the ages of one couple (at the end of the dance) leaves remainder R when divided by M, then every couple has to have sum of ages leaving remainder R when divided by M.

Given N, M, K and all of the dancers' ages, figure out in how many ways the dance can be performed. Since the dancers' ages are measured in seconds, the values can be very large.

Input

The first input line contains three integers N ($3 \leq N \leq 10^6$), M ($1 \leq M \leq 10^9$) and K ($1 \leq K \leq 10^9$), corresponding to the number of people in each circle, to the sacred number and to the number of dance steps, respectively.

The second input line contains N space-separated integers A_i ($1 \leq A_i \leq 10^9$), the ages of the women.

The second input line contains N space-separated integers B_i ($1 \leq B_i \leq 10^9$), the ages of the men.

Initially, the i-th man is aligned with the i-th woman. For each vector of ages, the first element is considered to be to the right of the last one.

Output

The output consists of a single integer, the remainder obtained by dividing the number of distinct dances by $10^9 + 7$.

Input example 1	**Output example 1**

`4 10 1`
`3 4 1 7`
`13 27 36 9`
	`1`

Input example 2	**Output example 2**

`5 10 2`
`3 4 1 7 6`
`4 7 1 2 5`
	`3`

Input example 3	**Output example 3**

`5 10 2`
`3 4 1 7 6`
`5 4 7 1 2`
	`4`

Input example 4	**Output example 4**

6 21 3
10 58 23 31 37 2
45 17 9 24 38 30
	42

Problem E
Party Company

Yankovich works as Software Engineer in a company called POI (Party Online Infinitely), as the name suggests it is a company that promotes online parties. To test the systems some employees threw parties and invited only the company employees with some restrictions.

The company employees form a hierarchical structure, where each one has a direct manager (except the company owner), and has no cyclic relations. Due to the company's promotion processes, the age of an employee is not greater than the age of his direct manager.

These initial parties work in the following way:

- Each party j has an owner O_j and an age range, from L_j to R_j.
- To be invited to a party an employee needs to have age inside the age range $[L_j, R_j]$ of the party. It is guaranteed that the age of the owner of each party is inside the age range of their parties.
- Besides the age restriction, to be invited to a party an employee must work directly with another employee that will be invited (as direct manager or direct subordinate), except for the party owner.

Yankovich is responsible for the application that computes data about the parties a user has participated. As an initial task he has to calculate how many parties an employee was invited to. Since he is late to deliver his first task, help him calculating this information for these initial parties.

Input

In the first line there are two integers N, M ($1 \leq N, M \leq 10^5$) representing the number of employees and the number of initial parties, respectively.

The next N lines contain the employee data. In the i-th of these lines there are two integers A_i, B_i ($1 \leq A_i \leq 10^5$, $1 \leq B_i \leq N$) representing the age of i-th employee and his direct manager (the employees are numbered from 1 to N, the company owner is the number 1 and he is the unique employee that has $B_i = i$). It is guaranteed that $A_i \leq A_{B_i}$.

The next M lines define the initial parties. In the j-th of these lines there are three integers O_j, L_j, R_j ($1 \leq L_j \leq A_{O_j} \leq R_j \leq 10^5$) representing the owner of this party, and its age range.

Output

Output one single line containing N integers (separated by a single space). The i-th of these numbers should be the number of parties the employee number i was invited to.

Input example 1

```
10 3
8 1
3 5
5 1
2 3
4 1
3 3
1 2
7 1
2 2
3 2
3 5 9
5 3 8
3 2 6
```

Output example 1

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

Problem F
Fastminton

The *Commission for the Regional deveLopment of Fastminton* (CRLF) organizes annual tournaments of the novel and unusual game of Fastminton, a Badminton derivative. CRLF is proud to be organizing the third big tournament, that will be held this year.

The commission's former programmer has developed a computerized system to capture and store the results from the matches, point by point, to individual files. He left before being able to complete a parser to the files, so the CRLF requires your help to guarantee that all the records can be read from the written files, to avoid loosing years of thrilling game results.

A summary of the Fastminton rules was given to you to help in your tasks. It is, in essence, a shorter (or faster) version of Badminton:

- Fastminton matches are always played with two opposing players in a court that is divided by a net;
- Players are identified by their relative positions on the score (left and right player);
- A match is composed of three games. The player that reaches 2 games is the winner;
- A game is won when a player reaches at least 5 points with a at least a 2 point difference from his opponent, or by reaching 10 point (whichever comes first);
- The left player begins serving on the first game. For all other games, the serving player is the one that won the last game;
- Every round results in a point for the server or the receiver player. The player who scores serves the next round.

Input

The input is comprised of a single line containing a sequence of characters. This line represents a complete match event sequence, and may contain the characters `S` (server point), `R` (receiver point) or `Q` (score announcement). The input does not contains consecutive score announcements.

Output

The program must print a line containing the current score for each score announcement (`Q`) found on the input.

If the game is underway, the announcement will be in the format "`GL (PL) - GR (PR)`", where `GL` and `GR` are the number of *games* won by the left and right players, and `PL` and `PR` are the current points of the left and right players. An asterisk (`*`) must be appended to the point marked of the player that will serve in the next round.

If the game has finished, the announcement will follow "`GL - GR`" with the string "`(winner)`" added at the end of the game marked of the winner player.

Input example 1	**Output example 1**

`SRSSQSSSSQRRSS`	`0 (1) - 0 (3*)`
	`0 (0) - 1 (2*)`

Input example 2	**Output example 2**

`SRSSQSSSSQRRSSQ`	`0 (1) - 0 (3*)`
	`0 (0) - 1 (2*)`
	`0 - 2 (winner)`

Input example 3	**Output example 3**
`RSRSSRRRRRRRRRRSSSSRRSQ`	`2 (winner) - 0`

Problem G

Game Show!

The Society of Bright Competitors (SBC) organizes television shows to its members (and currently it also broadcasts online!). SBC uses a a system of credits called *sbecs*, which can be used by players to participate in competitions or can be exchanged for prizes at the end of each season. SBC started a new type of game, and needs to do some simulations to avoid very large losses in the prize pool!

Ricardo is going to try the new game. He must bet 100 sbecs, which are transferred to his game balance. Then, a sequence of boxes is positioned. The game consists of rounds, and the maximum number of rounds is equal to the number of boxes. At each round, Ricardo decides whether to open the next box or to quit the game. If Ricardo quits, he gets the current balance of sbecs back. If Ricardo opens the next box, its content, which is a secret number, is added to his balance and the game continues. As the secret number in the box may be negative, Ricardo may end up at a loss! The game ends when Ricardo decides to quit or when the last box is opened.

SBC hired you to test the game. From the content of the boxes, you must decide what would be the largest possible balance that Ricardo could get.

Input

The first input line contains an integer C, $1 \leq C \leq 100$, which is the number of boxes in the game. After the first input line, there are C more lines. Each of the C lines contains the secret number of a box. The lines are in the same order of the boxes. The secret numbers are integers, V, $-1000 \leq V \leq 1000$.

Output

Output a line containing an integer which is the largest possible balance that Ricardo may get, given that sequence of boxes.

Input example 1	**Output example 1**
4 -1 -2 -3 -4	100

Input example 2	**Output example 2**
5 -10 20 -30 40 -50	120

Problem H
SBC's Hangar

A small cargo plane of the System of Binary Cargos (SBC) was designed to transport special and secret products. These products are grouped in boxes with different weights.

The plane has a safety weight range, within which the aircraft is stable. More specifically, there is an interval such that if the total weight of the boxes carried is not in the interval, then it will not be possible to guarantee the security of the flight.

It is known that all boxes have different weights. Moreover, given any two boxes, the heavier one weighs at least twice as much as the lighter box.

Your task is to determine in how many ways you can choose a specified number of these boxes to be transported on the plane without destabilizing it.

Input

The first line of the input contains two integers, N and K, representing respectively the number of available boxes and the specified number of boxes to be loaded in the plane.

The second line of the input contains N integers, separated by a space, and representing the weights of the boxes.

The third line of the entry contains two integers, A and B, which specify the weight safety range, which is the (closed) interval $[A, B]$.

Consider all weights reported in the same unit.

- $1 \leq N \leq 50$.
- $1 \leq K \leq 50$.
- the weight W of each box is in the interval $1 \leq W \leq 10^{18}$.
- $1 \leq A \leq B \leq 2 \times 10^{18}$.

Output

The output consists of a single line containing the number of possible ways to choose the specified number of boxes without jeopardizing the flight.

Input example 1	**Output example 1**

`3 2`
`10 1 3`
`4 13`
	`3`

Input example 2	**Output example 2**

`4 3`
`20 10 50 1`
`21 81`
	`4`

Input example 3	**Output example 3**

`6 3`
`14 70 3 1 6 31`
`10 74`
	`11`

Problem I
Interactivity

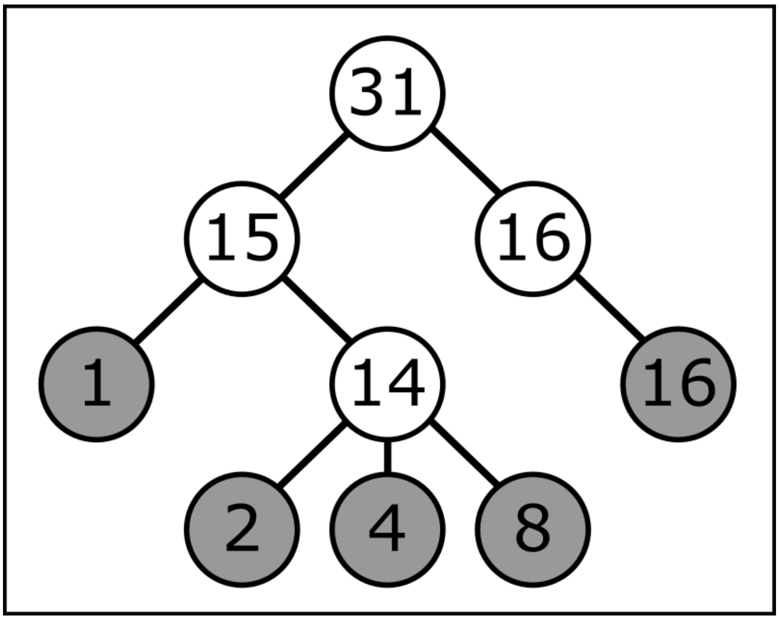
One day, Alice challenged Bob with the interactive programming problem described below.

≀≀

You have a tree (an acyclic connected graph). Each node of the tree has exactly one *parent*, except the *root* node, which has no parent. Nodes that are not parents of any nodes are called *leaves*. You know the structure of the tree, because you know the parent of each node that is not the root.

Each node contains an integer value. A node that is not a leaf contains the sum of the values of all its direct children. Thus, all the values in the tree are determined by the values contained in the leaves.

The picture below depicts an example. The leaves are marked as gray, while the other nodes are white. Each node shows the value it contains.



Initially, you don't know any of the node values, but you can query them one by one. Your task is to figure out the value of each node in the tree, making as few queries as possible.

≀≀

Bob solved this problem very easily. Then, to spice things up, Alice asked him: "given the tree structure, how many different solutions to this problem exist?" That is, how many different minimal sets of queries exist that allow one to figure out all the values stored in the tree? (Two sets of queries are considered different if and only if there is a node that is queried in one set but not in the other.)

Input

The tree has N nodes in total. Each node is identified by an integer between 1 and N, where node 1 is the root.

The input consists of two lines. The first line contains only the integer N.

The second line contains $N-1$ integers $P_1, P_2, \ldots, P_{N-1}$, separated by a single space, where P_i is the parent of node $i+1$, for $i = 1, 2, \ldots, N-1$.

$2 \leq N \leq 10^5$.

$1 \leq P_i \leq N$, for $i = 1, 2, \ldots, N-1$.

Output

The output consists of a single line, which must contain the number of different minimal solutions to the problem faced by Bob. As this number may be large, your answer must be calculated mod 1000000007 ($10^9 + 7$).

Input example 1	**Output example 1**

`3`
`1 1`
	`3`

Input example 2	**Output example 2**

`4`
`1 2 3`
	`4`

Input example 3	**Output example 3**

`5`
`1 2 2 2`
	`7`

Problem J
Collecting Data

Acre and Amanda are very curious and are always looking for patterns around them. They routinely collect and analyze data from various sources (city traffic, rainfall, number of leaves that fall from trees), hoping to find interesting patterns.

Their last expedition produced a very promising data set: it followed a perfectly straight line! Formally, it was a list of $N/2$ pairs of integers, possibly repeated. When these pairs were plotted as points in the Cartesian plane, all of them were perfectly collinear! Pleased, Acre and Amanda stored this data as a table containing the pairs of integers.

Unfortunately, while Acre and Amanda were out collecting more data, their infant son entered their office and scrambled the data table, by shuffling all the values in it. Now, all Acre and Amanda have left are the N shuffled integers. They want to try to reconstruct the original data table from them.

Formally, Acre and Amanda want to arrange these integers into $N/2$ pairs, where each pair represents a point in the Cartesian plane, in such a way that all these points are collinear. The given list of integers can have repeated values, and each value must be used exactly as many times as it appears in the list. The resulting data set can also have repeated data points.

Since there may be several different suitable datasets that can be formed with the given integers, Acre and Amanda want to know the number of such datasets. Two datasets are considered different if and only if there is a point that appears more times in one data set than in the other.

Input

The first line contains a single integer N, the length of the given list of integers. N is always even, because it is twice the amount of points from the original data set. The second line contains N integers, representing the list of values from the shuffled data table.

The integers are separated by a single space.

$4 \leq N \leq 200$.

Each value I from the list is in the interval $-10000 \leq I \leq 10000$.

Output

The output must consist of a single line with a single integer, the number of different ways the list of integers can be arranged into collinear points. Since this number can be very large, print the answer mod 1000000007 ($10^9 + 7$).

The answer can be zero for some cases.

Input example 1	**Output example 1**

`8`
`1 2 3 5 20 18 16 12`
	`2`

Input example 2	**Output example 2**

`6`
`1 2 3 4 5 20`
	`0`

Input example 3	**Output example 3**

`8`
`1 2 5 5 5 5 8 9`
	`10`

Problem K
Between Us

Ties are usually an issue on elections and games. Recently, a new game, called *Between Us*, was created. The game is played by a number of players over a social network. On each turn, there are multiple votings, where a player may receive only votes from friends. The player getting the largest number of votes wins the game.

The game is still in its design phase, but the game developers faced a very common issue: since the number of friends of a given player is usually small, ties are very common, which makes the game less fun. To avoid this situation, they decided to add a new module to the match-making system, where it will try to form groups of players such that each player has an odd number of friends in the same group.

This problems turned out to be more challenging than they expected and they are now focusing on a simpler variation: given a set of players, the module must find a *partition* of the players in at most two groups, satisfying the restriction that each player must have an odd number of friends in his group. The problem is that this partition is not always possible. Your task is to determine whether the partition is possible.

Input

The first line of input contains two integers, P and F, the number of players and the number of friendship relations, respectively, satisfying $2 \leq P \leq 100$ and $1 \leq F \leq P \times (P-1)/2$. Each of the next F lines will contain two integers A and B, with $1 \leq A, B \leq P$ and $A \neq B$, meaning that players A and B are friends. Each friendship relation is given at most once, that is, if a line contains A and B, no other line contains both numbers.

Output

Output a single line containing one character. If the partition is possible, then write the upper case letter '`Y`'; otherwise write the uppercase letter '`N`'.

Input example 1	**Output example 1**

`4 4`
`4 2`
`1 3`
`2 3`
`1 4`
	`Y`

Input example 2	**Output example 2**

`4 3`
`4 2`
`2 3`
`1 2`
	`Y`

Input example 3	**Output example 3**

`5 5`
`3 5`
`3 1`
`1 4`
`2 5`
`2 4`
	`N`

Problem L
Lavaspar

Word Search is a well-known hobby, although it is losing some of its prestige in recent years. The goal in this game is to find words in an array, where each cell in this matrix contains a letter.

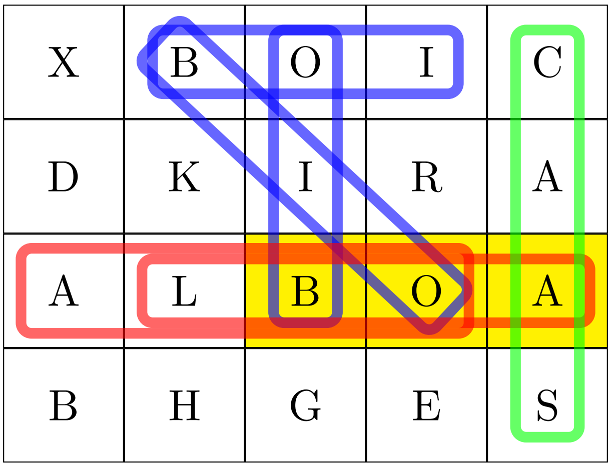
Bibika and her brother were playing Word Search, but soon they lost interest in the game, as finding all the words was getting relatively easy. Bibika would like to take her brother away from the computer, she searched the internet for games of the same style and ended up finding the *Lavaspar Hunting*.

Lavaspar Hunting is a game that follows the same idea of the famous Word Search. However, instead of simply having to find a word in the matrix, the goal is to find any anagram of the word, making the game more difficult and interesting. The anagram can be found in a row, column or diagonal.

An anagram is a word formed by rearranging the letters of another. Sometimes, an anagram does not exist as a word in the language, but this does not matter. BALO, LOBA and AOLB are examples of anagrams of the word BOLA.

Bibika realized that it was possible for the same cell in the matrix to make part of anagrams of different words and then she started to call these *special cells*.

Now she would like to know, given an array configuration and a collection of words, how many special cells are there?



X	B	O	I	C
D	K	I	R	A
A	L	B	O	A
B	H	G	E	S

The picture above illustrates the first example, where the collection of words consists of three words: BOLA, CASA and BOI. The rectangles of each color represent anagrams of different words from the entry. The 3 special cells are painted yellow.

Input

The first line contains two integers, L and C, which correspond to the number of lines and columns of the array, respectively.

Each one of the next L lines contains a word with C letters.

These lines are followed by a line containing an integer, N, which is the number of words in the collection of words to follow.

Finally, there are now N lines, each of which contains a word in the collection.

All characters in the array and in each word of the collection of words is a capital letter of the English alphabet.

No two of the N words in the collection are anagrams of each other.

- $2 \leq L$, $C \leq 40$.
- $2 \leq N \leq 20$.
- The number P of letters of each of the N words is in the interval $2 \leq P \leq \min(15, \max(L, C))$.

Output

The output consists of a single line that contains an integer corresponding to the number of special cells.

Input example 1	**Output example 1**

`4 5`
`XBOIC`
`DKIRA`
`ALBOA`
`BHGES`
`3`
`BOLA`
`CASA`
`BOI`
	`3`

Input example 2	**Output example 2**

`3 3`
`AAB`
`ABA`
`BAA`
`2`
`ABA`
`BBB`
	`3`

Input example 3	**Output example 3**

`2 4`
`AAAA`
`AAAA`
`2`
`AAA`
`BBB`
	`0`

Problem M
Machine Gun

Fulanito plays an old-school arcade game. In the game, he can place a machine gun anywhere within his base, which contains all (x, y) points with $x < 0$. There are N enemies on the battlefield. The i-th enemy ($1 \leq i \leq N$) is at position (x_i, y_i) with $x_i > 0$. All these positions are given in advance.

A machine gun at (x_m, y_m) covers an angle of vision to the right, centered on the line $y = y_m$, and whose borders are given by lines $y = y_m \pm \frac{x - x_m}{2}$. When placed, it kills all enemies within such angle, including its border lines.

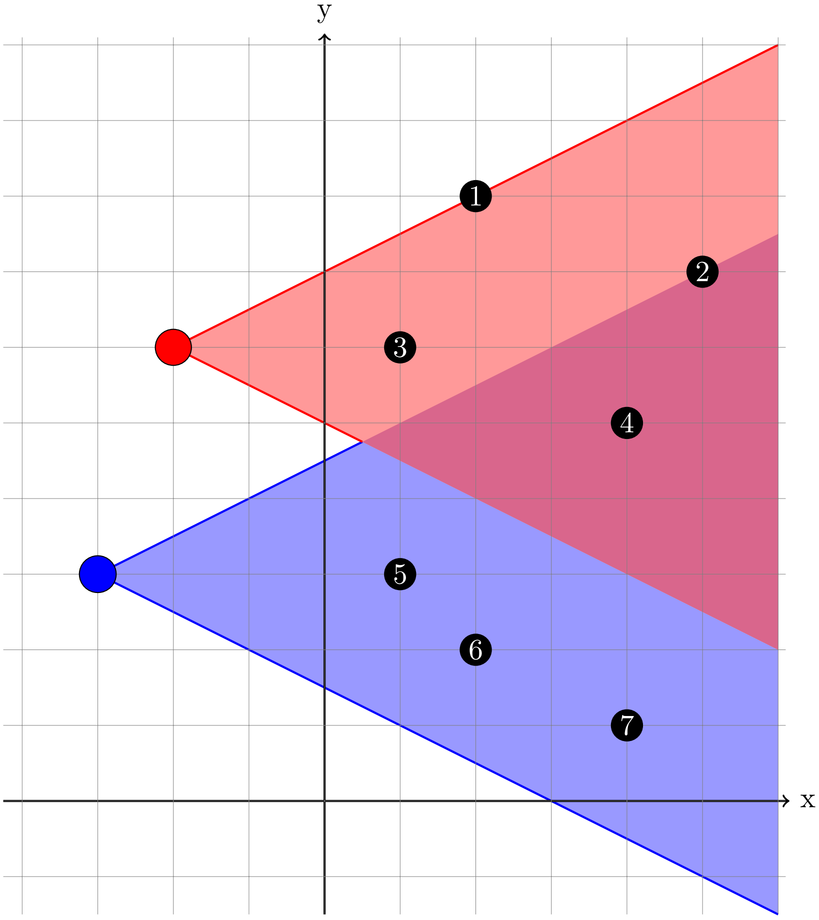


Figure 1: Pictorial representation of the sample input

The scoring system used by this videogame is extremely awkward: many believe that it was in fact a horrible bug by the game developers, who then proceeded to just yell "it's not a bug, it's a feature!" to anyone asking about it. Specifically, the score achieved by a certain placement of the machine gun is calculated by executing the following steps:

- Take the ids (i between 1 and N) of **all** the enemies that it kills.
- Sort all those ids in increasing order.
- Let the sorted values be $i_0 < i_1 < \cdots < i_{k-1}$
- Compute the score as $\left(\sum_{j=0}^{k-1} i_j \cdot 5782344^j\right) \mod (10^9 + 7)$

- Note: A machine gun that kills no enemies gets a score of exactly 0 points.

To get better at this game, Fulanito is going to ask you q questions: each question asks the score that would be achieved by placing the machine gun at a certain (x_m, y_m).

To make the problem more challenging, each (x_m, y_m) is not given directly. Instead, values a and b are given, such that $x_m = -1 - \big((p + a) \bmod (10^9 + 7)\big)$ and $y_m = (p + b) \bmod (10^9 + 7)$, where p is the answer to the previous query ($p = 0$ when processing the first query).

NOTE: It is guaranteed that in total, taking all queries together, the number of killed enemies is at most 10^6.

$1 \leq x_i, y_i \leq 10^9$

$1 \leq N, q \leq 10^5$

$0 \leq a, b < 10^9 + 7$

Input

The first line contains the two integers N, q.

The next N lines contains two integers each: x_i and y_i.

The next q lines contains two integers each: The values a and b that specify each query as explained in the problem statement.

Output

For each query, output a single line with a single integer containing the answer to that query.

Input example 1	**Output example 1**

`7 2`
`2 8`
`5 7`
`1 6`
`4 5`
`1 3`
`2 2`
`4 1`
`2 3`
`373785639 373785644`
	`626214369`
	`981053491`

Problem N
Number Multiplication

Eugenius is a very bright mathematician who enjoys multiplying numbers.

Once, he found M nodes written on a piece of paper, numbered from 1 to M, we call these M-nodes. Each node i was labeled with a unique prime number p_i, such that the primes were ordered (i.e. if $i < j$ then $p_i < p_j$).

Later on, he decided to draw N new nodes, called N-nodes, and add edges between nodes in M and nodes in N. He was very careful not to connect an M-node with an M-node, nor an N-node with an N-node; he wasn't as careful regarding how many edges between any two nodes he drew.

And just like that, he ended up with a bipartite multigraph.

As he mainly enjoyed multiplying numbers, he decided to label each N-node with the multiplication of all M-nodes that were directly connected to it; where each M-node was multiplied as many times as edges were between them.

Each N-node i ended up being labeled with a number c_i. Then, the following formula characterizes each c_i:

$$c_i = \prod_{(j,i) \in E} p_j,$$

where E is the multiset of added edges, and every edge e satisfies $e \in M \times N$. Later on, Eugenius decided to grab something to eat. While he was enjoying his torus, he accidentally spilled coffee on the labels p_i and they disappeared.

Can you help him recover those sorted prime numbers?

Input

The first line contains three integers M, N and K, the amount of M-nodes, the amount of N-nodes and the amount of *distinct* edges. These values are such that $1 \leq M, N < 10^3$ and $1 \leq K < 10^4$.

The next line has N numbers c_i (the labels of the N-nodes), such that $1 < c_i < 10^{15}$.

Finally, there are K lines, each of these has three numbers m n d, representing that there are d edges between M-node m and N-node n, satisfying $1 \leq m \leq M$, $1 \leq n \leq N$ and $1 \leq d \leq 50$.

It is guaranteed that all nodes (M-nodes and N-nodes) have degree at least 1.

Output

Output a single line with M sorted numbers, the prime labels of the M-nodes of indices $1, \ldots, M$ that keep Eugenius awake.

Input example 1

```
4 3 4
15 16 13
2 1 1
3 1 1
1 2 4
4 3 1
```

Output example 1

```
2 3 5 13
```

Input example 2

```
4 5 7
3 9 7 143 143
1 1 1
1 2 2
2 3 1
3 4 1
3 5 1
4 5 1
4 4 1
```

Output example 2

```
3 7 11 13
```

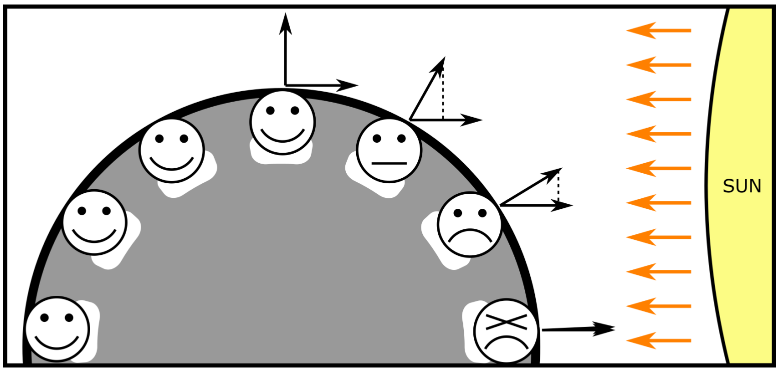
Problem O
Venusian Shuttle

The Human Colony at Venus is thriving! Here, the main form of public transportation is the Venusian Shuttle: it is a round flying disc, with windows and seats all around its border. In this shuttle, all the seats are window seats. And people are not allowed to move from one seat to another. So once someone chooses a seat, they must remain there until they leave the shuttle.

Even though it is a fully autonomous vehicle, every shuttle operates with an engineer on board, to handle any unexpected problems. You are Shuttle 1C9C's engineer, and you spend most of your work shift reading books. The problem is that you hate being exposed to the sun. Therefore, you want to choose a place to sit that minimizes the total amount of sunlight you take along your entire work shift.

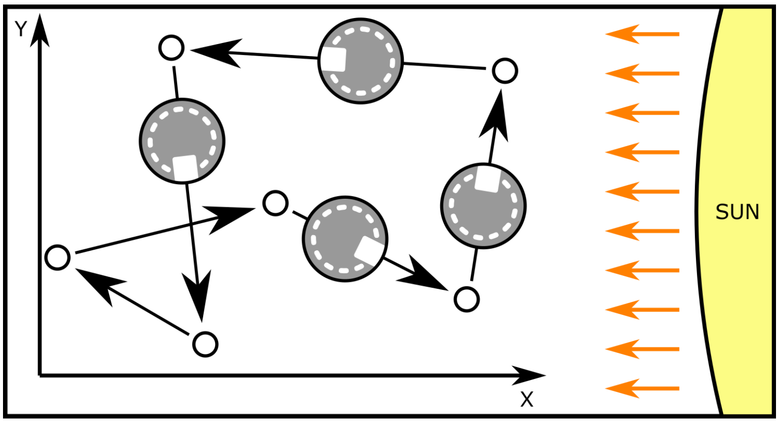
The colony is represented by the Cartesian plane, where the X axis points east and the Y axis points north. The days at Venus are very long (in fact, longer than the year!), so you can assume that the sun always shines from the east direction. That is, the sunlight always travels west, in the negative X direction.

See the figure below. The more your window is turned east, the more sunlight you're exposed to. But if your window is turned west, you're not exposed to the sun at all.



Formally, suppose the vector (D_x, D_y) represents the direction faced by your window. Note that you only get any sunlight if $D_x > 0$. And let θ be the angle between the vectors (D_x, D_y) and $(1, 0)$ (a vector pointing directly towards the sun). If $\cos(\theta) \leq 0$ then you don't get any sunlight. Otherwise, you receive $\cos(\theta)$ units of sunlight per second.

The shuttle route consists of a sequence of stations around the colony. The shuttle starts the work shift at the first station, visits all stations in order, and then returns to the first one.



The shuttle always follows a straight line when going from one station to another, moving with constant speed of one meter per second. And although the shuttle is round, it has a "front side": this side is always facing the direction it is moving forward, and the shuttle rotates accordingly when it changes direction at the stations.

You may ignore time spent turning the shuttle around, picking up or dropping off passengers.

Input

The first input line contains a single integer N, the number of stations visited during the shuttle's route.

Then, N lines follow, each line containing the coordinates X and Y of a station, separated by a single space.

The stations are given in the order they are visited.

Any station may be visited more than once in the route.

Any two consecutive stations are distinct, as well as the last and first stations.

All coordinates are given in meters.

$2 \leq N \leq 100000$.

The coordinates of each station are integers in the interval $-10000 \leq X, Y \leq 10000$.

Output

The output consists of a single line with a real number, the minimum total amount of sunlight you can take in a single tour around the shuttle route. Your answer should have exactly two digits after the decimal point.

Input example 1	**Output example 1**

`3`
`2 5`
`17 5`
`11 11`
	`6.00`

Input example 2	**Output example 2**

`4`
`3 0`
`3 6`
`6 3`
`0 3`
	`4.24`

Input example 3	**Output example 3**

`4`
`3 2`
`1 1`
`-3 -1`
`-1 0`
	`0.00`