مرکز مطالعات پارت
لذت کشف آینده

دانشگاه فردوسی مشهد

انجمن علمی گروه کامپیوتر

مسابقات برنامه‌نویسی دانشگاه فردوسی مشهد

FCPC

(طرح آموزشی)

ویژه عموم دانشجویان و دانش آموزان

۱۶ فروردین ۱۴۰۳

توضیحات:

- مسابقه شامل ۱۰ سوال در ۱۸ صفحه و زمان پاسخگویی ۴ ساعت است.
- کامپایلرهای موجود C, CPP, Java, Python هستند.
- ورودی‌ها و خروجی‌های تمام سوالات، ورودی/خروجی استاندارد است.

Question	Title
A	A_Landian families
B	Erfan and the keyboard
C	Ali the Gutty Boy
D	Amin vs. Emad
E	Morty, the number lover
F	Mrs. X and the Marco Polo
G	Saman and the Walls
H	Amin the Careless Driver
I	Good Formula
J	PWSCOFU

Problem A: A-Landian families

There was a problem in A-land Telecommunications and the internet disconnected. Therefore families decided to go to each other's houses and have little parties.

A-land is a very strange city. Every room in A-landian houses can contain at most m people. That's why two families are friends in A-land if and only if they can reserve some number of rooms and use their full capacity. For example, if every room in A-land can contain 5 people, then two families each with 4 and 8 people are not friends, while two families of sizes 5 and 10 are.

Hamed is the mathematician of the city, he wants to count the number of pairs of family friendships, but he doesn't know programming hence, cannot compute the answer. Help him find the number of family friendships in A-land.

Input (standard input)

The first line of input contains two integers n and m, the number of families in A-land and the room capacity.

The next line contains n integers a_i, the size of each family living in A-land. ($1 \le n,\ m \le 10^5$, $1 \le a_i \le 10^9$)

Output (standard output)

Print a single integer, number of family friendships in A-land.

Sample Input and Output

stdin	stdout
3 6 1 5 13	2
6 5 1 2 3 4 5 5	3

Problem B: Erfan and the keyboard

A-land has no internet for the next few hours. So Erfan has decided to watch some TV in the meantime.

In order to watch TV, he should find the remote. Finding the TV remote needs d_f calories of energy. After finding the remote he should enter the channel ID of each channel he wants to watch. Each channel ID is a sequence of digits 0-9. pressing the *i*-th digit needs di calories of energy. After entering the channel ID he should also press the “enter” key which needs d_{enter} calories of energy. Given the Channel IDs Erfan wants to watch, compute how much energy he needs.

Note that he has to find the remote only **once**.

Input (standard input)

The first line of input contains a single integer n, the number of channel IDs Erfan wants to watch. ($1 \leq n \leq 10^5$)

The second line of input contains two integers d_f and d_{enter}. ($0 \leq d_f, d_{enter} \leq 10^9$) The third line of input contains 10 integers d_i, the energy needed to press the *i-th* key.($0 \leq d_i \leq 10^9$) The fourth line contains *n* strings of digits s_i, the channel IDs Erfan wants to watch. ($|s_i| \leq 9$) Note that s_i might start or end in any digit.

Output (standard output)

On the only line of output print the amount of energy Erfan needs for watching TV .

Sample Input and Output

stdin	stdout

5
1 1
1 1 1 1 1 1 1 1 1 1
1 2 3 4 5
	11

1
1 1
1 1 1 1 1 1 1 1 1 1
12345
	7

Problem C: Ali the Gutty Boy

Ali has a special typing habit, he eats a peanut as for each time he presses a key in the keyboard. His team always has to bring a sack of peanuts for him when participating a programming competition. Now he has the task of typing the problems for FCPC.

As we are trying to minimize the amount of peanut Ali eats while typing FCPC problems, we have installed an auto-complete plugin on his typing program. The auto-complete plugin works like this: the plugin has an internal dictionary of words sorted by their frequency in the English language. Whenever a word is being typed, auto-complete suggests the most common word (if any) starting with all the letters typed so far. By pressing tab, the word being typed is completed with the auto-complete suggestion. Auto-complete can only be used after the first character of a word has been typed – it is not possible to press tab before having typed anything. If no dictionary word starts with the letters typed so far, pressing tab has no effect.

In order to minimize the amount of peanut consumed, we have realized that it is sometimes possible to use auto-complete to our advantage even when it is not suggesting the correct word, by deleting the end of the auto-complete word. For instance, to type the word “autocorrelation”, we start typing “aut”, which then auto-completes to “autocorrect” (because it is such a common word these days!) when pressing tab. By deleting the last two characters (“ct”) and then typing the six letters “lation”, the whole word can be typed using only 3 (“aut”) + 1 (tab) + 2 (backspace twice) + 6 (“lation”) = 12 keystrokes (only 12 peanuts to be fed to Ali!), 3 fewer than typing “autocorrelation” without using auto-complete.

Given the dictionary on the auto-complete plugin and the words Ali wants to type, output the minimum number of peanuts required to type each word. The only keys Ali can use are the letter keys, tab and backspace.

Input (standard input)

The first line of input contains two positive integers *n* , the number of words in the dictionary, and *m*, the number of words to type. Then follow *n* lines with one word per line, sorted in decreasing order of how common the word is (the first word is the most common). No word appears twice in the dictionary. Then follow *m* lines, containing the words to type. The dictionary and the words to type only use lower case letters 'a'-'z'. The total size of the input file is at most 1 MB.

Output (standard output)

For each word to type, output a line containing the minimum number of keystrokes (peanuts!) required to type the corresponding word.

Sample Input and Output

Stdin	stdout
8 6 prog program xyz horses lango languages kanguages kango program xyz noprefix horse language kanguage	4 2 8 3 6 3

Problem D: Amin vs. Emad

The problem-setting committee has had a lot of problems recently. The worst one was working in a secret place so that no one would be able to know the problems before the contest. That's why they changed their place every day.

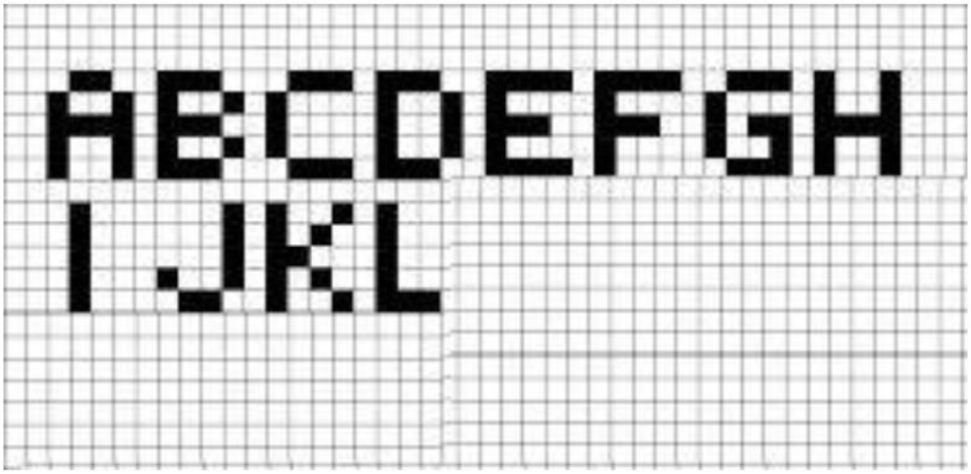
At the beginning of everyday Erfan writes down the encrypted name of the lab and everyone else just reads the small piece of paper and goes to the specified lab. The paper used is an $n \times m$ table where each cell in the table is either empty or contains a character from "A" to "L".

But one day Emad arrived at the university just after Erfan and saw the piece of paper. He, unable to decrypt the word, first changed some empty cells to character cells and then torn the paper apart.

Some minutes later, Amin came and saw the pieces of paper. He could guess what happened. Amin knows that the word on the paper was either s_1 or s_2 or ... or s_q. Now he wants to know how many of the candidate words can possibly be the correct word.

He took a picture from the pieces of paper and sent it to you to help him.

Each character has a special pattern, shown below. Given the picture Amin sent you, find the answer to his question.



For making the problem a little easier, each character can only be made with each cell of the pattern being equal to that character. For example, there is an "A" in a table, if and only if, there is a pattern like below in the table.

	A	A	
A			A
A	A	A	A
A			A
A			A

Also, there are only characters from “A”, ”B”,…, ”L” in the input. Note that a pattern like below counts as two “A”s.

	A	A		A	A	
A			A			A
A	A	A	A	A	A	A
A			A			A
A			A			A

Also, note that patterns can not be rotated or mirrored i.e. the patterns below do no represent “J”:

J			
J			
J			
J			J
	J	J	

	J			
J				
J				
	J	J	J	J

Input (Standard Input)

The first line of the input contains three integers *n, m* and *q*, The number of rows and columns in the input and the number of candidate words. ($1 \leq n, m \leq 300, 1 \leq q \leq 10^5$) The next *n* lines denote the picture Amin sent you. Each character in the table is either a “.” (representing an empty cell) or an uppercase letter from “A” to “L”. The next *q* lines each contain a single string with uppercase alphabet. It’s guaranteed that input will not contain more than 10^6 characters.

Output (standard output)

In the only line of output print the number of candidate strings that can be formed from the letters in the input table

Sample Input and Output

stdin	stdout

`10 10 3`
`BBB.......`
`B..B......`
`BBB.......`
`B..B......`
`BBB.......`
`......AA..`
`.....A..A.`
`.....AAAA.`
`.....A..A.`
`.....A..A.`
`AB`
`BA`
`BABA`
	`2`

Problem E: Morty, the number lover

Morty loves to play with numbers. He has even named some numbers with special names! He calls a number x, d-good if the following is satisfied:

1.d is a digit, i.e. $0 \leq d \leq 9$

2.suppose $x = x_0 + x_1 \times 10^1 + x_2 \times 10^2 + \cdots + x_n \times 10^n$ such that $0 \leq x_i \leq 9$ and $x_n \neq 0$ then for each i such that $x_i = d$, i should be odd.

For example, 172, 180 and 7070 are all 7-good numbers while 777 and 17 are not 7-good. One day Morty came up with the following problem:

How many numbers n exist such that:

1.$l \leq n \leq r$

2.n is d-good for a given d

3.n is divisible by m

Help him find the answer modulo $10^9 + 7$.

Input (standard input)

The only line of input consists of 4 space-separated integers: l, r, d, m ($1 \leq l \leq r \leq 10^{2000}$, $0 \leq d \leq 9$, $1 \leq m \leq 2000$)

Output (standard output)

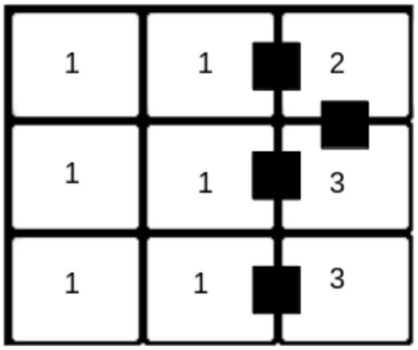
In the only line of output print the answer to Morty's problem modulo $10^9 + 7$

Sample Input and Output

Stdin	stdout
1 10 3 2	5
1 12 3 2	6
1 12 2 2	4

Problem F: Saman and the Walls

After getting accepted in the driving license exam and becoming a police officer, Saman has been selected as the head of the engineering faculty of a well-known university. In his first day, he found out that the main building of the faculty has the shape of a tabular network (like the figure below) and there are doors between any two cells sharing a side. So, logically, he decided to replace some doors with walls. But he should always know how many groups of cells there are such that it's possible to go from one to another using the doors.



For example the table above could be the state after 4 actions, the lines with black squares are walls while other lines are doors. In the figure above there are 3 groups of connected cells. The number in each cell indicates its group number.
As Saman is only a police officer and a philosopher he has asked you to help him find the number of such groups after each action he orders.

Input (standard input)

The first line of input contains three integers *n*, *m*, and *q*, representing the number of rows and columns the network has and the number of actions Saman is going to take. ($1 \leq n, m \leq 10^3$; $1 \leq q \leq min(maximum\_number\_of\_walls, 10^6)$).

The next *q* lines contain the information about each action in chronological order. Each line has the form: "d r c" which d is either "up", "down", "right" or "left" and $1 \leq r \leq n$, $1 \leq c \leq m$.

The action "d r c" means that Saman wants to put a wall between the cell located at *r*-th row and *c*-th column and its neighbor at side *d*.

It's guaranteed that no door will be in the input twice. It's also guaranteed that each action will remove the wall between two existing cells e.g. "up 1 1" cannot be in the input because there is no cell above cell (1,1).

Output (standard output)

The output should contain exactly *q* lines.

On the *i*-th line print the number of groups of cells mentioned in the statement after the *i*-th action.

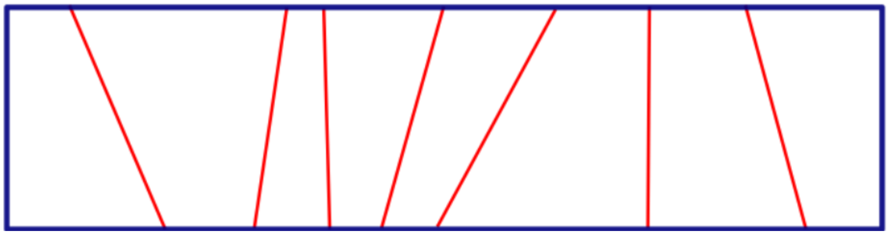
Sample Input and Output

Stdin	Stdout
3 3 4 left 1 3 down 1 3 right 2 2 left 3 3	1 2 2 3
4 4 4 up 2 2 right 2 2 down 2 2 left 2 2	1 1 1 2

Problem G: Mrs. X and the Marco Polo

Mrs. X has just come back from her journey around Europe. Not many know yet but she is preparing for another trip to Portugal in the near future, leaving Mozambique the last country in the world she has not visited yet.

She is now collecting and preparing her baggage. Mrs. X's suitcase consists of multiple cells separated by walls (see the figure below). She has a disciplined and precise procedure for placing the goods in the suitcase: throwing them as randomly as possible towards the cells between the walls.



Knowing the landing position of each thrown good and the walls, help Mrs. X to see how many goods she has shot in each cell. She needs to know this as the airlines have a limit on the number of goods in each cell!

Input (standard input)

The input consists of multiple test cases. The first line of each test case contains six integers n, m, x_1, y_1, x_2, y_2. n is the number of walls, m is the number of goods, (x_1, y_1) is the coordinate of the upper-left corner of the suitcase, and (x_2, y_2) is the coordinate of the lower-right corner of the suitcase. ($0 \leq x_1, x_2, y_1, y_2 \leq 10^9$)

In each of the next n lines, there are two numbers U_i, L_i representing the x-coordinate of the upper and lower points of the i-th wall. It's guaranteed that the walls do not intersect and $x_1 \leq U_i$, $L_i \leq x_2$. It's also guaranteed that U_is and L_is are given in increasing order.

Finally, in each of the next m lines there are two integers *Xi*, *Yi* representing the coordinate of the landing position of i-th the good.

$(0 \leq n, m \leq 10^5, x_1 \leq X_i \leq x_2, y_2 \leq Y_i \leq y_1)$

You can assume that the goods won't land on the walls, borders, and outside of the suitcase. The input terminates with a single line containing 0.

Output (standard output)

For each cell of the suitcase from left to right, first print the number of the cell (starting from 0 to n) and the number of the goods in the cell should be printed in a single line in the format stated in the samples. Print an empty line between the outputs of consecutive test cases.

Sample Input and Output

stdin	stdout

5 6 0 10 60 0
3 1
4 3
6 8
10 10
15 30
1 5
2 1
2 8
5 5
40 10
7 9
4 10 0 10 100 0
20 20
40 40
60 60
80 80
5 10
15 10
25 10
35 10
45 10
55 10
65 10
75 10
85 10
95 10
0
	0: 2
	1: 1
	2: 1
	3: 1
	4: 0
	5: 1
	
	0: 2
	1: 2
	2: 2
	3: 2
	4: 2

Problem H: Amin the Careless Driver

Part software group was officially found in July 2013 in order to play the role of an intelligent economic assistant for its clients, but the problem has nothing to do with it. We just had to name the sponsor.

Amin is well known for his careless driving, however, He himself, as the writer of this problem, does not agree for sure. He believes that he never breaks a rule, and he is right, but the fact is that the driving rules are a bit different in Mashhad. Amin has recently decided to change his attitude and driving habits according to the international rules -or at least the national ones. He wants to start by understanding the mechanism of speed cameras. Help him with that!

There are n speed cameras across Vakil-abad which is a straight highway.x_1, x_2, … , x_n represent the distance of each camera from the beginning of Vakil-abad in meters. The speed of all vehicles across Vakil-abad must not exceed V meters per minute or they will be sentenced a speeding ticket.

The mechanism of the cameras is as follows: each vehicle starts its journey from the beginning of Vakil-abad and keeps going until the end. When a vehicle passes a camera, the camera saves the current time in terms of minutes. Finally, when the vehicle finished its journey, it will be fined according to the number of pairs of speed cameras between which its average speed has strictly exceeded the speed limit. You are to write a program to compute this number for Amin's car.

Input (standard input)

The input comes in three lines.

The first line contains two space separated integers n, V indicating the number of speed cameras and the maximum legal speed of vehicles along the highway. $(1 \leq n \leq 10^5; 1 \leq V \leq 10^9)$

The second line contains n integers $x_1, x_2, \ldots, x_n$ indicating the distance of each camera from the beginning of the highway in meters, sorted in ascending order i.e. $0 \leq x_1 < x_2 < x_3 < \ldots < x_n \leq 10^9$.

Finally, the third line contains n numbers $t_1 < t_2 < \ldots < t_n$. The i-th number shows the minute which the vehicle passes the i-th camera. $(0 \leq t_i \leq 10^9)$

Output (standard output)

Print in a single line the number of pairs of speed cameras between which the vehicle has exceeded the speed limit.

Sample Input and Output

stdin	stdout

3 1
1 3 5
1 3 5
	0

3 1
1 3 5
1 3 4
	2

Problem I: Good Formula

Salam You are given a sequence of N non-negative integers: A_1, A_2, …, A_N

Consider inserting a + or - between each pair of adjacent terms to make one formula.

There are 2^{N-1} such ways to make a formula. Such a formula is called **good** when the following condition is satisfied:

- - (mines) does not occur twice or more in a row.

Find the sum of the evaluations of all good formulae. We can prove that this sum is always a non-negative integer, so print it modulo $10^9 + 7$

Input (standard input)

First line contains N. Next line contains N space separated integers, A_1, A_2, …, A_N.

($1 \leq N \leq 10^5, 1 \leq A_i \leq 10^9$)

Output (standard output)’

Print the sum modulo $10^9 + 7$.

Sample Input and Output

Stdin	stdout
3 3 1 5	15
4 1 1 1 1	10
10 866111664 178537096 844917655 218662351 383133839 231371336 353498483 865935868 472381277 579910117	279919144

As an example for the second sample we have the following five good formulae:

- 1 + 1 + 1 + 1 = 4
- 1 + 1 + 1 – 1 = 2
- 1 + 1 – 1 + 1 = 2
- 1 – 1 + 1 + 1 =2
- 1 – 1 + 1 – 1=0

Thus, the answer is 4 + 2 + 2 + 2 + 0 = 10

Problem J: PWSCOFU

Pardis Water Sports Complex of Ferdowsi University(PSWSCOFU) has N lifeguards, each of which has a shift that covers some contiguous interval of time during the day. For simplicity, the pool is open from time t = 0 until time t = 10^9 on a daily basis, so each shift can be described by two integers, giving the time at which a lifeguard starts and ends his shift. For example, a lifeguard starting at time t = 4 and ending at time t = 7 covers three units of time (note that the endpoints are "points" in time).

Unfortunately, the university has hired 1 more lifeguard than it has the funds to support. Given that it must fire exactly one lifeguard, what is the maximum amount of time that can still be covered by the shifts of the remaining lifeguards? An interval of time is covered if at least one lifeguard is present

Input (standard input)

The first line of input contains N($1 \leq N \leq 10^5$). Each of the next N line describes a lifeguard in terms of two integers in the range 0 … 10^9, giving the starting and ending point of a lifeguard's shift.

All such endpoint are distinct. Shifts of different lifeguards might overlap.

Output (standard output)

Write a single number, giving the maximum amount of time that can still be covered if university fires 1 lifeguard.

Sample Input and Output

Stdin	stdout

3
5 9
1 4
3 7
	7