



## Gran Premio de México 2022 - Primera Fecha

This problem set is used in simultaneous contests:  
Primera Fecha Gran Premio de México 2022  
Primera Fecha Gran Premio de Centroamérica 2022

*May 14th, 2022*

### Problems book

#### General Information

This problem set contains 12 problems; pages are numbered from 1 to 14, Without considering this page. Please, verify your book is complete.

#### A) About your program

1) The code submitted to resolve a problem should be named : *problem\_code.c*, *problem\_code.cpp*, *problem\_code.java* or *problem\_code.py*, *problem\_code.kt*, where *problem\_code* is the uppercase letter that identifies the problem. Remember that in Java the name of the main class should be the same as the name of the file.

#### B) About Input

- 1) Your program should read the input from *standard input*.
- 2) When an input line contains more than one value, these values are separated by a single white space. The input does not contain any other white spaces.
- 3) Each line of input, including the last one contains exactly only one end of line character.
- 4) End of input is on the end of file.

#### C) About output

- 1) The output of your program should be written to *standard output*.
- 2) When an output line contains more than one value, these values should be separated by a white space. The output does not contain any other white space.
- 3) Each line of output, including the last one, should contain exactly one end of line character.

## Problem A

### Anya's Gifts



Anya has a big problem. Her father's birthday and National Father's Day happen in the same week. Anya loves her father so she bought a large number of gifts to give him. Anya doesn't want to give him all gifts in one day, that's why she will give some gifts to her father on his birthday and the rest of them on father's day. Of course, all gifts must be given and her father must receive at least one gift on each of the two dates.

Anya's father is a very elegant man with refined tastes, so his gifts should be as elegant as possible. At the time of purchasing the gifts, the store owner gave an elegance score to each of the gifts. Anya wants to distribute the gifts in such a way that the sum of the XOR of the elegance score of the gifts she gives on Father's Day and the XOR of the elegance score of the gifts she gives on his birthday is maximum.

Help Anya accomplish this difficult and elegant task.

#### Input

The first line contains an integer  $N$  ( $2 \leq N \leq 10^5$ ) indicating the number of gifts Anya bought.

The next line contains  $N$  integers  $e_i$  ( $0 \leq e_i < 2^{50}$ ) for  $i = 1, 2, \dots, N$ , denoting the score of elegance of each present.

#### Output

Output a single line with an integer representing the result of this task.

<b>Input example 1</b> 4 1 2 4 8	<b>Output example 1</b> 15
<b>Input example 2</b> 4 10 8 6 17	<b>Output example 2</b> 41

## Problem B

# Building 5G Antennas

Treeland is a country consisting of  $n$  cities and  $n - 1$  bidirectional roads. As you might imagine, Treeland is a tree, which means that there is exactly one simple path between each pair of cities.

The president of Treeland plans to build a 5G network in the country during  $n$  days. Everyday, a 5G antenna tower will be built in a different city according to the following rules:

- Each day, an antenna must be built in a city at a distance not greater than  $k$  from a city with an antenna already built. This restriction does not apply for day 1.
- If during  $i$ -th day there are multiple valid cities to build an antenna, the one with the smallest number must be chosen.

More formally, let  $P = [p_1, p_2, \dots, p_n]$  be a permutation where  $p_i$  is the city where an antenna is built during day  $i$ . For all  $i > 1$  there must be a  $j < i$  such that  $\text{dist}(p_i, p_j) \leq k$ , and  $P$  must be the lexicographically smallest possible permutation. Here we define  $\text{dist}(p_i, p_j)$  as the number of roads in the simple path from  $p_i$  to  $p_j$ .

Find and print  $P$ .

### Input

The first line contains two integers  $n$  and  $k$  ( $1 \leq n \leq 10^5$  and  $1 \leq k \leq 100$ ).

Each of the following  $n - 1$  lines contains two integers  $u$  and  $v$ , indicating that there is a road connecting cities  $u$  and  $v$ .

### Output

Print a single line with  $n$  integers separated by a space — the answer to the problem.

<b>Input example 1</b> 3 1 1 3 2 3	<b>Output example 1</b> 1 3 2
<b>Input example 2</b> 5 2 1 4 1 5 4 2 5 3	<b>Output example 2</b> 1 2 3 4 5
<b>Input example 3</b> 5 1 1 2 1 5 2 4 3 5	<b>Output example 3</b> 1 2 4 5 3

## Problem C

# Candies Median

You are the teacher of a kindergarten classroom and there will be a party in a few days. As expected, you are in charge of bringing all the candies for the kids. You have created  $q$  possible plans over the  $n$  types of candies that exist. For each plan:

First of all, the  $i$ -th type of candy has a sweetness level of  $a_i$  (this value can be negative, in the case of sour or spicy ones). You have sorted all these values in ascending order ( $a_i < a_{i+1}$  for all valid  $i$ ).

Then, you choose  $k$  tuples  $(l, r, x)$  that mean that you'll buy  $x$  candies of each type with id in the range  $[l, r]$ .

The *beauty* of a plan is the value of the median of all the sweetness levels of the candies you choose. Your task is to compute the beauty of each plan.

### Input

The first line of input contains two integers  $n$  and  $q$  ( $1 \leq n, q \leq 5 \cdot 10^5$ ) — The number of types of candies and the number of plans to analyze.

The second line of input contains  $n$  integers  $a_i$  ( $|a_i| \leq 10^9$ ,  $a_i < a_{i+1}$  for all valid  $i$ ) — The sweetness level of the types of candy.

The lines describe  $q$  plans.

The first line of a plan contains an integer  $k_i$ , the number of tuples of the  $i$ -th plan.

The following  $k_i$  lines contain three integers  $l_j$ ,  $r_j$  and  $x_j$  ( $1 \leq l_j \leq r_j \leq n$ ,  $1 \leq x_j \leq 10^6$ ) — The limits of the range of ids to choose and the number of candies to choose from each type, respectively.

It is guaranteed that the sum of  $k_i$  among all the plans doesn't exceed  $5 \cdot 10^5$ .

### Output

Print  $q$  lines — The  $i$ -th line must contain the beauty of the  $i$ -th plan. Your answer will be considered correct if its relative or absolute value doesn't exceed  $10^{-9}$ .

In other words, let's assume that your answer is  $a$ , and the answer of the jury is  $b$ . The checker program will consider your answer correct, if  $\frac{|a-b|}{\max(1,b)} \leq 10^{-9}$

Input example 1	Output example 1
7 3	9
-19 -2 0 9 17 18 20	0
2	0
4 6 1	
2 6 4	
1	
3 3 1	
2	
3 4 2	
1 5 1	

Input example 2	Output example 2
7 3	0.5
-15 -9 -7 0 1 12 15	15
1	6.5
2 7 1	
1	
7 7 1	
4	
5 6 1	
3 6 5	
6 7 4	
7 7 2	

## Problem D

# Different Pass a Ports

Francovich is planning to travel the world via boat, he insists on being via boat because he really loves marine life and marine food.

Each time Francovich arrives at a port, the travel agent at the port will stamp in his pass-a-port that keeps track of the arrival for legal purposes, at the first page that has not yet being stamped, each page of the pass-a-port can have at most one stamp, and the port will stamp the pass-a-port even if Francovich has arrived to that port in his travel.

Francovich noticed that the pass-a-port will be different depending on the order that he visits the ports, and because he doesn't have a clear destination and want to pass by ports as many times as he can because he doesn't like direct travel, Francovich wonders what is the different numbers of ways the stamps on his pass-a-port can look and is your task to help him!

The ports have predefined travel routes that are bidirectional, and Francovich only has enough money to do exactly  $K$  travels and starts in port number 1.

### Input

The first line will have three integers  $N$ ,  $M$ , and  $K$  ( $2 \leq N \leq 100$ ,  $1 \leq M \leq 5000$ ,  $1 \leq K \leq 10^5$ ) that describe the number of ports, the number of travel routes between ports, and the number of travels that Francovich can do.

The next  $M$  lines will have two integers  $A$ ,  $B$  ( $1 \leq A, B \leq N$ ) each that describe a direct travel route between the port  $A$  and the port  $B$

### Output

A single integer  $X$  that describes the number of ways Francovich's pass-a-port can look after his  $K$  travels. Because this number can be very large print it modulo  $10^9 + 7$ .

<b>Input example 1</b> 6 5 1 1 2 1 3 3 6 1 4 1 5	<b>Output example 1</b> 4
<b>Input example 2</b> 6 5 2 1 2 1 3 3 6 1 4 1 5	<b>Output example 2</b> 5

Input example 3	Output example 3
6 5 4 1 2 1 3 3 6 1 4 1 5	22

## Problem E

# Erudite Of Words

Ivanovich is highly considered an Erudite of words.

But Ivanovich doesn't know what it means, maybe is about knowing a lot of words, and the ability to use them in the right context is not important.

So Crystalovich wanted to test how good of an Erudite is Ivanovich, so she asked him how many words there are with length  $N$  using an alphabet of  $M$  different letters.

Francovich hearing the problem thought it was very easy, and he doesn't like easy problems, so he added a condition such that each word have to use exactly  $K$  different letters.

Help Ivanovich answer the question, since the answer can be very large print it modulo  $10^9 + 7$

### Input

The first and only one line contains three integers  $N$ ,  $M$  and  $K$  with  $(1 \leq N \leq 10^6)$   $(1 \leq K \leq M \leq 5 \times 10^3)$  representing the length of the word, the size of the alphabet, and the number of different letters that the words should have.

### Output

A single number indicating the numbers of words modulo  $10^9 + 7$

<b>Input example 1</b> 3 3 3	<b>Output example 1</b> 6
<b>Input example 2</b> 5 2 2	<b>Output example 2</b> 30
<b>Input example 3</b> 1000 1000 1	<b>Output example 3</b> 1000



## Problem F

# Froginald the Frog

Froginald is a very happy frog living in his pond, he got very excited when he knew that the Grand Prix of Mexico is starting again so he decided to go immediately to participate! Even if that means missing an exam, a class, or a day of work.

The road for the Grand Prix is filled with rocks, rocks that we need to overcome to be closer and closer to the Grand Prix.

In the case of Froginald, he can simply jump over them, the road of rocks is a linear path to the Grand Prix, and Froginald can jump to  $+1$  of his position or  $+2$ , always ending up closer to the goal. Unfortunate for Froginald some of the rocks are missing being that if he tries to jump to a place with a missing rock he would end up drowning in the lake, so he would definitely try to avoid that.

The road to the Grand Prix is long, and there are many ways to travel it, Froginald is so excited about starting his journey that he wants to know how many ways is for him to reach the goal!

Froginald starts in position 0 which will always have a rock, as well as position  $N$ .

### Input

The first line has two integers  $N$  and  $M$  ( $1 \leq N \leq 10^9$ ) ( $1 \leq M \leq 1000000$ ) indicating how long the road is, and how many holes are in the road. The next line will have  $M$  integers indicating that the position  $a_i$  doesn't have a rock.

### Output

A single integer that represents the number of ways that Froginald has to reach the position  $N$ , because this number can be very large print it modulo  $10^9 + 7$ .

Input example 1	Output example 1
5 1 2	2

## Problem G

# Going to the Regional

The Grand Prix of Mexico wishes to organize its next regional event in the same venue for all the participants, also this year it has a budget to help the participants to move to the venue of the event, so it wants to calculate how much transportation would cost per participant towards headquarters. They also want the process to be as comfortable and easy as possible for the participants, because they want them to stress as little as possible before the day of the regional, so each participant will be picked up at their door.

The organization in Mexico does not have enough vehicles to do this, so it has contacted a transport office to calculate the costs and they have given a cost per kilometer for each participant depending on the location of their home. The transport office ensures it can provide a vehicle for each participant and that will start the travel simultaneously for all participants to the desired venue.

The organization plans to give an equal budget to each participant for logistical reasons when making the final calculation of expenses, so given the cost per kilometer per participant and the exact location of each participant, can you help calculate the minimum cost per participant such that all participants can meet in one place?

For simplicity the distance from a participant to any other location is calculated in Euclidean distance.

### Input

In the first line of input a number  $n$  ( $1 \leq n \leq 100$ ) the number of participants

In the next  $n$  lines three integers  $x_i$  ( $-10^4 \leq x_i \leq 10^4$ ),  $y_i$  ( $-10^4 \leq y_i \leq 10^4$ ) and  $c_i$  ( $1 \leq c_i \leq 10^4$ ) the pair  $(x_i, y_i)$  are the coordinates of the participant  $i$  and  $c_i$  is the cost per kilometer that it takes to transport the participant  $i$

### Output

In a single line print the minimum budget per participant the organization has to invest to take all the participants to a single place

Your answer is considered correctly if its absolute or relative error does not exceed  $10^{-4}$

Formally let your answer be  $a$ , and the jury's answer be  $b$ . Your answer is accepted if and only if  $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-4}$

<b>Input example 1</b> 2 1 1 1 -1 -1 1	<b>Output example 1</b> 1.414214
<b>Input example 2</b> 4 0 1 1 1 0 1 0 -1 1 -1 0 1	<b>Output example 2</b> 1.000001

## Problem H

# Hog Fencing

Jaime has been successful building fences to keep his sheep safe, the fences he builds are very classic wooden fences forming a rectangle or a square. Seeing the success he has with sheep, he plans on expanding his business and will build fences for hogs. He built the first fence for some hogs and after a couple days, the fence was broken, hogs are really clever to find a way out of the fence.

Jaime has developed a new fence that uses an unbreakable metal wire, he will build a fence and verify the hogs are not able to escape from it, nor break it. To build the fence Jaime has  $N$  units of this metal wire, each unit with a 1 meter size. Since the metal wire is unbreakable he has to build each of the 4 walls for the fence using complete metal wire units and then join them at their ending points forming the rectangular fence. Note the metal wire units can not be broken, and a wall containing  $L$  metal wire units will have  $L$  meters in length.

Jaime has a lot of hogs now, and no fence ready, this is why he wants to build the fence that encloses the maximum possible area he can using this new metal wire. Jaime is not really good at math, so he asked your help to find the maximum area he can enclose building his fence with the  $N$  units he has.

### Input

The first and only line of input contains a single integer  $N$  ( $4 \leq N \leq 1000$ ). The number of metal wire units Jaime has to build the fence.

### Output

Print a line with an integer, representing the maximum area Jaime can enclose with a fence built using the  $N$  units.

<b>Input example 1</b> 4	<b>Output example 1</b> 1
<b>Input example 2</b> 15	<b>Output example 2</b> 12

## Problem I

### Isabel's Divisions

Isabel is a very curious girl. She is learning to count and to divide numbers, as she is a beginner performing divisions she can divide only by a one digit number, for example, she can divide 123456 by 2 but not by 20. To train her division skills she invented a simple game, she writes a number  $N$  of at most 8 digits and then she counts how many digits that form  $N$  divide  $N$  without reminder, for example, if  $N = 111$ , each of the three 1 divide  $N$  and then she counts 3. if  $N = 39$  the only digit in  $N$  that divides  $N$  is 3 and then she counts 1.

Isabel wants you to play with her, but you are busy training for the next programming contest, then you decided to make a program that obtains the count Isabel will get for each  $N$ . This is your task in this problem.

#### Input

The input consists of a single line, which contains a positive integer number  $N$  of at most 8 digits.

#### Output

Your program must output a single line, containing an integer, representing the count Isabel will get for the number  $N$ .

<b>Input example 1</b> 111	<b>Output example 1</b> 3
<b>Input example 2</b> 39	<b>Output example 2</b> 1
<b>Input example 3</b> 39333000	<b>Output example 3</b> 4
<b>Input example 4</b> 12345678	<b>Output example 4</b> 4

## Problem J

# Jeffrey's Ambition

Jeffrey is the richest man in the world. In fact, he is so rich that he could buy all the major companies himself. Ordinary people don't like this, they wish every rich man was limited to having only one company.

The world council has listened to the requests of the people and has put the most important companies in the world up for sale. Each of the  $N$  rich men in the world have given a list of the companies they would like to buy. The world council will be in charge of deciding which company will sell to each of the richest men.

Unfortunately, by meeting these conditions, it is possible for some companies to become ownerless. Which is terrible, since it is possible that Jeffrey take advantage of it and try to buy them, being thus, owners of more than one company.

People do not want this to happen, that is why the world council has given you the task of deciding which company to sell to each rich man, in such a way that at the end of the sale the minimum number of companies ends up without an owner.

Remember that rich men will only buy companies that are on their list.

### Input

The first line contains two integers  $N$  and  $M$  ( $1 \leq N, M \leq 10^4$ ) indicating respectively the number of rich men and the number of major companies.

Each of the next  $N$  lines describe each rich man list of companies. Each line starts with an integer  $K$  ( $0 \leq K \leq M$ ) indicating the number of companies he would like to buy, followed by  $K$  integers  $c_i$  ( $1 \leq c_i \leq M$ ) for  $i = 1, 2..K$  indicating the company he would like to buy.

It is guaranteed that the sum of  $K$  does not exceed  $10^5$ .

### Output

Output a single line with an integer indicating the minimum number of companies that will end up without an owner.

<b>Input example 1</b> 5 6 2 1 2 0 1 3 1 4 2 1 5	<b>Output example 1</b> 2
<b>Input example 2</b> 5 5 1 1 1 2 1 3 1 4 1 5	<b>Output example 2</b> 0

## Problem K

### Kilo Waste

A shelter for homeless people wants to buy  $n$  kilograms of rice for their dinner, but the market only offers bags of rice in predefined weights, making hard for the shelter to buy rice without waste.

For example, if the shelter wanted to buy 10 kilos of rice for the dinner and the available presentations at the market have bags of rice with 2, 15, and 7 kilos, the shelter can buy exactly 10 kilos buying 5 bags of 2 kilos and no rice will be wasted that night. However, if the shelter wanted to buy 5 kilos, there is no way they can buy rice in a way there is no waste, in this case, the minimum waste they can achieve is 1 (buying 3 bags of 2 kilos).

Given a list of rice sales presentations available at the market, and the amount of rice the shelter wants to buy for each of the next  $k$  dinners, find the minimum rice waste the shelter can have for each dinner.

#### Input

The first line of input contains two integer numbers separated by a space  $k$ , and  $p$ , ( $1 \leq k \leq 10^5$ ,  $1 \leq p \leq 50$ ) representing the number  $k$  of dinners the shelter will make, and the number  $p$  of presentations of rice the market sells. The second line of input contains  $p$  integer numbers separated by space, where the  $i$ -th number represents the amount of rice  $r_i$  ( $1 \leq r_i \leq 100$ ) on the  $i$ -th rice presentation available at the market. Each of the next  $k$  lines contains an integer number, where the  $i$ -th line represents the amount of rice  $n_i$  ( $1 \leq n_i \leq 5 \times 10^4$ ) the shelter needs to buy for the  $i$ -th dinner.

#### Output

Print  $k$  lines, where the  $i$ -th line contains a single integer number, the minimum rice waste the shelter can have after buying rice for the  $i$ -th dinner.

<b>Input example 1</b> 2 3 2 15 7 10 5	<b>Output example 1</b> 0 1
<b>Input example 2</b> 2 3 11 24 35 105 10	<b>Output example 2</b> 0 1
<b>Input example 3</b> 3 2 10 17 1 16 27	<b>Output example 3</b> 9 1 0

## Problem L

### Last Problem

The Leones(0,0,0) have finally retired from competition after several years in college and have decided to take a nice vacation at the beach. Although they are a little sad to leave their years of competition in the Grand Prix of Mexico.

They were relaxing in an enramada, ordering some piña coladas and looking at the sea in the distance, when a group of sailboats began to line up on the horizon... it just so happened that the sailboats had numbered sails. After a few minutes the great sage of the Caribbean arrived (also known as the man who sells prepared coconuts) and proposed a riddle in which they would win free coconuts if they solved it correctly. The team could not refuse to do one last problem and they got ready to solve it.

The riddle said the following: Visualize all the sailboats on the horizon as a list of numbers  $A$  then take an element  $A_i$  and calculate the sum of elements for all possible ranges  $[l, r]$  such that  $A_i$  has the maximum value among all elements between  $A_l$  and  $A_r$  inclusive.

This riddle was clearly very simple, so after the team answered it correctly, the wise man from the Caribbean decided to make the riddle even more interesting. Now, for the prize of free coconuts for life, the riddle said: let's say that the previous riddle is the result of  $f(i)$  for a sailboat at position  $i$ , then solve the sum of  $f(i)$  for all existing sailboats

This puzzle has exceeded the capabilities of the participants and now they ask for help from the next generations of ICPC

#### Input

In the first line a number  $n(1 \leq n \leq 10^6)$  the number of sailboats in the sea.

In the second line  $n$  integers that represents the numbers in the sails of the boats.  $(-10^9 \leq a_i \leq 10^9)$

#### Output

The first and only line in the output is the answer to the riddle modulo  $10^9 + 7$

<b>Input example 1</b> 4 1 2 5 3	<b>Output example 1</b> 58
<b>Input example 2</b> 7 1 2 3 4 6 5 1	<b>Output example 2</b> 297



## Gran Premio de México 2022 - Segunda Fecha

This problem set is used in simultaneous contests:  
Segunda Fecha Gran Premio de México 2022  
Segunda Fecha Gran Premio de Centroamérica 2022

*September 24th, 2022*

### Problems book

#### General Information

This problem set contains 12 problems; pages are numbered from 1 to 14, Without considering this page. Please, verify your book is complete.

#### A) About your program

1) The code submitted to resolve a problem should be named : *problem\_code.c*, *problem\_code.cpp*, *problem\_code.java* or *problem\_code.py*, *problem\_code.kt*, where *problem\_code* is the uppercase letter that identifies the problem. Remember that in Java the name of the main class should be the same as the name of the file.

#### B) About Input

- 1) Your program should read the input from *standard input*.
- 2) When an input line contains more than one value, these values are separated by a single white space. The input does not contain any other white spaces.
- 3) Each line of input, including the last one contains exactly only one end of line character.
- 4) End of input is on the end of file.

#### C) About output

- 1) The output of your program should be written to *standard output*.
- 2) When an output line contains more than one value, these values should be separated by a white space. The output does not contain any other white space.
- 3) Each line of output, including the last one, should contain exactly one end of line character.



## Problem A

# Advanced Player Setup

Ivanovich is obsessed with a new game, Algorithmic Creed, here you work with your creed trying to give free will to the coders of the world, meanwhile in the other side the managers, also known as “templars” want to take away coders free will and deprive them of their human nature.

An important part of this game is to obtain and update items to be a stronger coder, like a gaming mouse, a new keyboard, a better monitor... because you know that the most important part of coding is the setup.

The game has  $N$  items numbered from 1 to  $N$  and the only way to obtain new items is performing a trade. The only item you can buy directly is a pen (for writing pseudocode in paper) which is always represented with the number 1, you can always buy it for 1 coin and you can buy it several times. In this game you have the ability to trade items for another one which is always better, the game has a list of item trading rules, each rule describe the item  $u$  as a requirement to obtain item  $v$  and the number of  $c$  coins you have to give to perform the trade. In order to obtain item  $v$  you need to trade all items  $u$  that are a requirement for  $v$  and pay the number of coins associated to trade  $u$  for  $v$ , after the trade you will lose the  $u$  items, and the coins, but obtain item  $v$ .

Ivanovich has never been good at games, since he always loses focus and begins to think about programming problems. He is wondering how many different setups he can build in the game if he has  $C$  coins to trade. Can you help Ivanovich to solve this weird problem so he can go back to the game?

A setup is different from another one if there is at least one different item. A setup must have at least one item.

### Input

The first line of input contains three integers separated by a space  $N$  ( $1 \leq N \leq 10^5$ ),  $M$  ( $1 \leq M \leq 10^5$ ),  $C$  ( $1 \leq C \leq 10^2$ ), representing the number of items in the game, the number of trading rules available and the number of coins available respectively.

Each of the next  $M$  lines contains three integers separated by a space,  $u_i$  ( $1 \leq u_i \leq N$ ),  $v_i$  ( $1 \leq v_i \leq N$ ),  $c_i$  ( $1 \leq c_i \leq C$ ), indicating that in the  $i$ -th trading rule the item  $u_i$  is a requirement for  $v_i$  paying  $c_i$  coins.

### Output

In the first and only line, print the number of different setups Ivanovich can build. As the answer might be very large, please output the answer modulo  $10^9 + 7$ .

<b>Input example 1</b> 3 2 4 1 2 1 2 3 1	<b>Output example 1</b> 10
<b>Input example 2</b> 3 3 4 1 2 1 1 3 1 2 3 1	<b>Output example 2</b> 8

## Problem B

# Binahuatls Prophecy

The year 2012 started a trend when everybody tried to predict the end of the world, they always find something peculiar in the number that indicates that it will really be the end of the world.

One famous group was an old Mexican tribe called the Binahuatls, who predicted that the world will be ending in a year where the binary representation it's a palindrome, meaning that the number it's the same if we were to reading forward or backward.

But as you, the person sitting in a chair, reading this problem in a contest or in practice, that is thinking about what the question of this problem is, are a living example that the world hasn't ended yet. The most eager fans of the Binahuatl defend that the world will end in a binary palindromic year for sure.

Given two dates  $A$  and  $B$  help those eager fans to know how many binary palindromic years exists between  $A$  and  $B$  inclusive!

### Input

The first line contains a single integer  $Q$  ( $1 \leq Q \leq 10^5$ ), representing the number of test cases.

Each of the following  $Q$  lines contains two integer numbers separated by a space  $A$  and  $B$  ( $1 \leq A, B < 2^{31}$ ).

Output a line with a single integer that represents the closest palindromic year to  $X.L$

### Output

Print  $Q$  lines, where the  $i$ -th line contains a single integer number representing the answer for the  $i$ -th test case from the input.

<b>Input example 1</b> 1 1 10	<b>Output example 1</b> 5
<b>Input example 2</b> 3 10 10 4 4 2 2	<b>Output example 2</b> 0 0 0
<b>Input example 3</b> 4 123 1234 2000 2022 10 100 100 1000	<b>Output example 3</b> 47 1 14 42

## Problem C

# Correcting School Enrollment Errors

Planet E-13 orbits around a star in a faraway galaxy named UAZ. In that planet, everyone attends the same super powerful university, which has the capacity to receive up to  $N$  students per semester. Every student gets a school ID, and at the beginning of each semester, each of the students request to enroll in up to eight courses. School enrollment applications are registered by the secretary of the school department, however, given that sometimes she must do it very quickly because the deadlines are very close, students could end up enrolled in courses that they did not request and/or they could have courses they requested to be enrolled in but they were not. Given the list of students with their course requests, and the list of courses in which they actually ended up enrolled in the school system, your job is to provide a list of changes to be made to the school system's records, so that effectively the students end up enrolled in each and every one of the courses they requested, without having missing or unrequested courses.

### Input

In the first line of input there will be two integers  $N$  and  $M$  ( $1 \leq N, M \leq 10^5$ ), the number of students requesting at least one course enrollment this semester and the number of students that got at least one course enrollment this semester, respectively.

The next  $N$  lines, start with two integers  $A$  ( $1 \leq A \leq 10^7$ ) and  $B$  ( $1 \leq B \leq 8$ ), the ID of the student and the number of class enrollments student  $A$  is requesting respectively. Then in the same line will be  $B$  integers  $b_i$  ( $1 \leq b_i \leq 10^4$ ), this are the enrollment requests by student  $A$ .

The next  $M$  lines, start with two integers  $A$  ( $1 \leq A \leq 10^7$ ) and  $C$  ( $1 \leq C \leq 8$ ), the ID of the student and the number of class enrollments student  $A$  received. Then in the same line will be  $C$  integers  $c_i$  ( $1 \leq c_i \leq 10^4$ ), this are the enrollments assigned to the student  $A$  by the school.

note: students and enrollment lists will appear in any order.

### Output

If there was no error in enrollments, you should print "GREAT WORK! NO MISTAKES FOUND!" in one line, without quotes.

Otherwise you need to print the changes needed in enrollment such that all students get exactly the courses they requested, not more, not less.

For each student that needs a change in their enrollment there will be a line: the ID of the student followed by a list of corrections separated by spaces, a correction is of type  $-m$  or  $+m$ , which are removing a course and adding a course respectively. The ID's of students should appear in ascending order, also corrections should be ordered in ascending order for each student.

#### Input example 1

```
3 3
1 4 5 4 2 10
4 3 1 2 11
3 2 9 2
4 3 1 2 11
1 4 5 4 2 10
3 2 9 2
```

#### Output example 1

```
GREAT WORK! NO MISTAKES FOUND!
```

**Input example 2**

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

**Output example 2**

```
1 -1 -2 -3 -5
2 -2 -8 -9
3 -1 -3
4 +10
5 -2 -4 +8
8 -2 -4 -5
9 +7 +8
MISTAKES IN 7 STUDENTS: 14 NOT REQUESTED, 4 MISSED
```

## Problem D

# District Capitalism Roads

You are the CEO of a big company that makes roads in the city, one day you are commissioned to make roads in the district capitalism, and of course, the names clearly tell you, that district capitalism is a district where they really love the capital city! You have seen the city and you love the city too so you decided to work for free, it's a normal thing in the capital city that everybody works for free.

There are  $N$  cities numbered from 1 to  $N$ , you have a list of  $M$  roads that you can make, each road option is represented by three integers  $A$ ,  $B$ , and  $C$  which means that there is an option to connect city  $A$  to city  $B$  with cost  $C$ . Since everybody loves the capital city there is a fee for emotional damage that is charged to make a road away from the capital to  $Y$ : a road that has a distance  $X$  from the capital will cost  $X * C$ ,  $X$  is calculated by the number of roads in a simple path that exists between capital and  $Y$ . The capital city has number 1.

Calculate the minimum amount of money you need to spend to connect every city to the capital such that the distance from each city to the capital is minimal.

How can you pay for these roads if you are working for free? Why everybody loves the capital? Those are not the questions in this problem!

### Input

The first line, you will have two integers  $N$  ( $1 \leq N \leq 10^5$ ) and  $M$  ( $1 \leq M \leq 10^5$ ) representing the number of cities and the number of roads respectively. Each of the next  $M$  lines contains three integers  $A$ ,  $B$  ( $1 \leq A, B \leq N$ ) and  $C$  ( $1 \leq C \leq 10^5$ ) describing each of the road options you can possibly make.

### Output

Output a line with a single integer that represents the minimum cost it will take to connect all the cities to the capital, two cities are considered connected if there is a set of roads that you can use to reach from one city to another, it guaranteed that a solution exists.

<b>Input example 1</b> 5 4 1 2 10 1 3 4 1 4 12 1 5 1	<b>Output example 1</b> 27
<b>Input example 2</b> 5 5 1 2 10 1 3 4 2 3 8 5 4 12 1 5 1	<b>Output example 2</b> 39

Input example 3	Output example 3
5 4 1 2 10 2 3 4 3 4 12 4 5 1	58

## Problem E

# Express Warehouse Migration

Jaime's distribution company is getting back to business. In this new distribution task Jaime has to move  $N$  boxes that are stored in his backup warehouse to the main warehouse. In order to do this, Jaime has only one truck, so he will drive to the backup warehouse, then load the truck, and finally drive to the main warehouse and unload the truck, all the previous process is considered a single trip of boxes. Jaime knows if he loads more than  $M$  boxes in the truck it will force the engine and potentially break it, so he will ensure to move the boxes without forcing the engine.

Can you help Jaime determine, how many trips does he has to make in order to move all the boxes from the backup warehouse to the main warehouse?

### Input

The first and only line of input contains two integers separated by a space,  $N$  and  $M$  ( $1 \leq N, M \leq 10^6$ )

### Output

Print a single line with an integer number, the minimum number of trips Jaime has to make to move all the boxes to the warehouse.

<b>Input example 1</b> 10 10	<b>Output example 1</b> 1
<b>Input example 2</b> 1 10	<b>Output example 2</b> 1
<b>Input example 3</b> 11 5	<b>Output example 3</b> 3

## Problem F

# Famous Paintings

You are one of the most famous painters, Vincent Van Gogh and now you are thinking of painting another masterpiece. Everyone knows that your most famous paintings always contain geometric figures, specifically figures with exactly 4 sides, the interpretation of a simple figure in so many colors and in so many contexts is what made you the most famous painter.

Now, you are in front of your canvas painting your soon-to-be-famous starry night. In your canvas, there are already  $N$  stars and you want to connect 4 of these stars to show a quadrilateral constellation. In how many ways you can choose these 4 stars to make a quadrilateral figure?

### Input

In the first line, you will have an integer  $N$  ( $1 \leq N \leq 100$ ) that represents the number of stars on the canvas.

Each of the next  $N$  lines contains two integer  $X$ , and  $Y$  separated by a space ( $0 \leq X, Y \leq 10^6$ ) that represent the coordinates of the  $a_i$  star, no two stars share the same point.

### Output

Output a single line with an integer indicating the total ways you can choose 4 stars to draw your quadrilateral figure.

<b>Input example 1</b> 5 10 10 10 20 20 20 20 10 25 15	<b>Output example 1</b> 5
<b>Input example 2</b> 6 1 1 2 2 3 3 5 10 20 10 20 5	<b>Output example 2</b> 21
<b>Input example 3</b> 7 2 8 9 4 3 1 38 43 10 11 49 31 30 20	<b>Output example 3</b> 47



## Problem G

# Guadalajara Trains

In Guadalajara, the most recent train line is the Pi Line. The name is because it's after the third line and before the fourth line.

This line has two train rails and  $N$  stations, one rail is for a train that goes from station 1 to station  $N$ , and the other one is for a train that goes from station  $N$  to station 1. At each station, there is a given time for the train to wait to pick people up that can vary from station to station. Both trains have the same speed (1km/s) and the distance between each station is known. JP began is wondering how long both of the trains will be waiting in the same station?

Knowing the distance between consecutive stations, the time that the trains wait at each station, and the fact that the two trains begin to work at the same time, help JP to answer his question!

### Input

The first line contains a single integer  $N$  ( $1 \leq N \leq 10^6$ ) representing the number of stations in the Pi line. The next line contains  $N - 1$  integers separated by a space where the  $a_i$  ( $1 \leq a_i \leq 10^6$ ) integer represents the distance between the station  $i$  and the station  $i + 1$ . The third and last line of input contains  $N$  integers separated by a space where the  $x_i$  ( $1 \leq x_i \leq 10^6$ ) integer represents the time that the train will be waiting at station  $i$ .

### Output

Print a line with a single integer that indicates the time the two trains are waiting in the same station.

<b>Input example 1</b> 3 1 1 1 10 1	<b>Output example 1</b> 10
<b>Input example 2</b> 3 1 5 1 10 1	<b>Output example 2</b> 6
<b>Input example 3</b> 6 20 13 10 15 20 10 21 5 10 20 7	<b>Output example 3</b> 0

## Problem H

# How Many Laughs

When it comes to numbers there are two rules that are common knowledge:

- Some numbers are really funny.
- Not all people consider the same number as funny.

For example, some people find the number 24 funny, while other people assures the number 25 is even funnier.

As with some other topics there are some funny number jokes that only experts on the topic find entertaining: if your funny number is 25 and your friend says 100, you find it funny as even when 100 is not your funny number, it is disguised! If you divide 100 by 4 it becomes your funny number!.

You are a funny numbers clown and will be presenting a funny numbers show tonight. This means you want to make people laugh with funny number jokes. As you want to be the best funny numbers clown in the city you do not want to expose yourself trying too hard to make people laugh, so you will not say a number greater than  $X$ . You will perform in front of your  $N$  friends, all your friends are funny number experts and find entertaining a funny number joke. Because you are very close to each of your friends you know what is the number each of them finds funny.

Help yourself to know how many different numbers you can say tonight to make at least one of your friends laugh!

### Input

The first line of input contains two integers separated by a space  $N$  and  $X$  ( $1 \leq N \leq 20$ ,  $1 \leq X \leq 10^9$ ) indicating respectively the number of friends in the show and the maximum number you can say.

The second and last line contains  $N$  numbers separated by a space  $a_i$  ( $1 \leq a_i \leq 10^9$ ) where the  $i$ -th number represent the funny number of your  $i$ -th friend.

### Output

Output a single line with an integer, indicating how many different numbers you can say tonight to make at least one of your friends laugh.

<b>Input example 1</b> 2 10 2 3	<b>Output example 1</b> 7
<b>Input example 2</b> 3 100 2 3 5	<b>Output example 2</b> 74
<b>Input example 3</b> 1 500 5	<b>Output example 3</b> 100

## Problem I

# Inversion Counting

You have a sequence of  $N$  integers  $a_i$  with elements between 1 and  $K$ , and you want to calculate the number of inversions. To make it more complicated you get  $Q$  operations  $i$ , change all occurrences of  $i$  to  $i + 1$ , and vice versa.

Each operation changes the original sequence for the following operations.

**note:** is considered an inversion to a pair of indices  $(i, j)$ , where  $i < j$  and  $a_i > a_j$ .

### Input

The first line contains three integers  $N, K, Q$  ( $1 \leq K \leq N \leq 100\,000, 1 \leq Q \leq 1\,000\,000$ ).

The next line contains  $N$  integers  $a_1, a_2, \dots, a_N$  ( $1 \leq a_i \leq K$ ) specifying the sequence.

The following  $Q$  lines each contain an integer  $i$  ( $1 \leq i \leq K - 1$ ), representing the operation of swapping  $i$  elements with  $i + 1$ , and vice versa.

### Output

For each  $Q$  operation, print a single integer, the number of inversions as specified in the statement.

Input example 1	Output example 1
5 4 3	4
1 4 2 1 2	2
3	2
2	
1	

## Problem J

# Joining the KAK

The KAK (Klub Algorithmic Kukei) is a club from the Kukei university which has a peculiar story about how they got their name: One time they made advertisement and some merch that had their name printed, after everything was paid they noticed that they misspelled the word Club as Klub, they find it easier to keep the name rather than making all the advertisement and merch again.

Because of this experience now the KAK focuses a lot on grammar, that is why if you want to join the KAK they will ask something related to strings.

Today you want to join, and in your evaluation for joining the KAK they gave a lot of cards with letters in them, each card has exactly one letter. They ask you to find given a value a  $k$  what would be the  $k$ -th string if you sorted lexicographically all the different strings that you can make arranging at least one of the cards and at most all of them? Two arrangements  $A$  and  $B$  are different if they have a different amount of cards or if at some position  $x$  the letters of the cards at position  $x$  differ.

Gain your entry to the “Klub” answering the question!

### Input

The first line of input contains two integers  $N$  and  $K$  ( $1 \leq N \leq 1000$ ,  $1 \leq K \leq 10^6$ ) representing the number of cards that the KAK gave you and  $k$ . It is guaranteed that at least  $k$  different strings can be made with the given cards.

The next line contains a string of size  $N$  representing the  $N$  pieces. All the letters are lowercase English letters.

### Output

Print a line containing a single word, the answer to the problem to join the KAK.

<b>Input example 1</b> 3 2 aaa	<b>Output example 1</b> aa
<b>Input example 2</b> 3 15 abc	<b>Output example 2</b> cba
<b>Input example 3</b> 10 120873 abcdefghij	<b>Output example 3</b> fgdhic

## Problem K

# Krystalova's Trivial Problem

Krystalova is writing trivial problems (as always) but now for some reason (lucky for you) she also wrote a trivial statement.

You have a trivial list of numbers  $L$ , and two trivial operations:

1.  $u\ l\ r\ x$ : Add  $x$  to the numbers in  $L$  in the range  $[l, r]$ .
2.  $q\ l\ r$ : Get the sum of  $L_i^2$  for every  $i$  in the range  $[l, r]$ .

### Input

In the first line of input two integers  $N$  ( $1 \leq N \leq 10^5$ ),  $Q$ , the size of  $L$  and the number of operations respectively.

The next line contains  $N$  integers  $L_i$  ( $0 \leq L_i \leq 10^8$ ), the elements of  $L$ .

The next  $Q$  lines contains an operation, that can be one of the two types described before. If the line starts with a letter ' $u$ ' then will be three integers  $l$  ( $1 \leq l \leq N$ ),  $r$  ( $1 \leq r \leq N$ ),  $x$  ( $-10^8 \leq x \leq 10^8$ ). If the line starts with a letter ' $q$ ' then will be two integers  $l$  ( $1 \leq l \leq N$ ),  $r$  ( $1 \leq r \leq N$ )

### Output

For every operation of type 2, print in one line the result of the operation. As the answer might be very large, please output the answer modulo  $10^9 + 7$

Input example 1	Output example 1
5 5	14
1 1 1 1 1	9
u 1 3 1	25
q 1 5	
u 3 5 1	
q 3 3	
q 1 5	

## Problem L

# Limited Increasing Sequences

Krystalova is studying a very particular sequence that she likes very much. Krystalova named the sequence: “Limited Increasing Sequence”. A sequence  $X$  of positive integer numbers is a limited increasing sequence if and only if every element  $X_i$  is not greater than the sum of all the items  $X_j \mid j < i$ . In other words, every element  $X_i$  meets the following condition:

$$\sum_{j=1}^{i-1} X_j \geq X_i$$

Let  $f(X)$  be the sum of all the elements in a limited increasing sequence, it is said the limited increasing sequence  $X$  produces  $K$  if  $f(X) = K$ .

Looking at this sequence, Krystalova found that a number  $K$  may be produced as the result of different  $X$ . She wants to know how many different limited increasing sequences produces  $K$ .

Help Krystalova to solve her problem!

### Input

In the first line, you will have a single integer  $K$  ( $1 \leq K \leq 10^6$ )

### Output

Print a line with a single number with the result. As the answer might be very large, please output the answer modulo  $10^9 + 7$

<b>Input example 1</b> 10	<b>Output example 1</b> 84
<b>Input example 2</b> 2022	<b>Output example 2</b> 904964280
<b>Input example 3</b> 3	<b>Output example 3</b> 1

## Maratón de programación de la SBC 2022

El set de problemas de este cuaderno es utilizado simultáneamente en las siguientes competencias:

Maratona SBC de Programação 2022  
Tercera Fecha Gran Premio de México 2022  
Gran Premio de Centroamérica 2022  
Torneo Argentino de Programación 2022  
The 2022 ICPC Bolivia Preliminary Contest

*08 de Octubre de 2022*

### Cuaderno de Problemas

#### Información General

Este cuaderno contiene 14 problemas; Las páginas están numeradas de 1 a 19, sin contar esta página. Verifique que su cuaderno está completo.

#### A) Sobre los nombres de los programas

- 1) Para soluciones en C/C++ y Python, el nombre del archivo de código fuente no es significativo, puede ser cualquier nombre.
- 2) Si su solución es en Java, el archivo debe ser llamado: `codigo_de_problema.java` donde `codigo_de_problema` es la letra mayúscula que identifica al problema. Recuerde que en Java el nombre de la clase principal debe ser igual que el nombre del archivo.
- 3) Si su solución es en Kotlin, el archivo debe ser llamado: `codigo_de_problema.kt` donde `codigo_de_problema` es la letra mayúscula que identifica al problema. Recuerde que en Kotlin el nombre de la clase principal debe ser llamado igual que el nombre del archivo

#### B) Sobre la entrada

- 1) La entrada de su programa debe ser leída de *entrada standard*.
- 2) La entrada está compuesta de un único caso de prueba, descrito en un número de línea que depende del problema.
- 3) Cuando una línea de entrada contiene varios valores, estos están separados por un único espacio en blanco; la entrada no contiene ningún otro espacio en blanco.
- 4) Cada línea, incluyendo la última, contiene exactamente un caracter de final-de-línea.
- 5) El final de la entrada coincide con el final del archivo.

#### C) Sobre la salida

- 1) La salida de su programa debe ser escrita en *salida standard*.
- 2) Cuando una línea de salida contiene varios valores, estos deben ser separados por un único espacio en blanco; la salida no debe contener ningún otro espacio en blanco.
- 3) Cada línea, incluyendo la última, debe contener exactamente un caracter de final-de-línea.

Promocional:



Sociedade Brasileira de Computação





## Problema A

## Acertando los Monótonos No Triviales Maximales

En este problema trataremos con secuencias de caracteres, comúnmente llamadas *strings*. Una secuencia es *no-trivial* si contiene al menos dos elementos.

Dada una secuencia  $s$ , decimos que el tramo  $s_i, \dots, s_j$  es *monotono* si todos sus caracteres son iguales, y decimos que es *maximal* si el tramo no puede ser extendido hacia la izquierda o la derecha perdiendo monotonicidad.

Dada una cadena compuesta únicamente por los caracteres “a” y “b”, determine cuántas veces aparece el caracter “a” en tramos monotonos maximales no-triviales.

**Entrada**

La entrada consiste de dos líneas. La primer línea contiene un solo entero  $N$ , donde  $1 \leq N \leq 10^5$ . La segunda línea contiene una cadena una cadena con exactamente  $N$  caracteres, que consta únicamente de los caracteres “a” y “b”.

**Salida**

Imprima una sola línea conteniendo un entero, representando el número de veces que aparece el caracter “a” en tramos monotonos maximales no-triviales.

<b>Ejemplo de entrada 1</b> 7 abababa	<b>Ejemplo de salida 1</b> 0
<b>Ejemplo de entrada 2</b> 7 bababab	<b>Ejemplo de salida 2</b> 0
<b>Ejemplo de entrada 3</b> 10 aababaaabb	<b>Ejemplo de salida 3</b> 5
<b>Ejemplo de entrada 4</b> 10 bbaababaaa	<b>Ejemplo de salida 4</b> 5

## Problema B

# Brincando piedras

Alice y Bob jugarán un juego con 3 montones de piedras. Los jugadores se turnan y, en cada turno, un jugador debe elegir un montón que todavía tiene piedras y quitarle un número positivo de piedras. Gana quien toma la última piedra del último montón que aún tenía piedras. Alice hace el primer movimiento.

El  $i$ -ésimo montón tendrá un número aleatorio y uniformemente distribuido de piedras en el rango  $[L_i, R_i]$ . ¿Cuál es la probabilidad de que Alice gane dado que ambos juegan de manera óptima?

### Entrada

La entrada consta de una línea con 6 enteros, respectivamente,  $L_1, R_1, L_2, R_2, L_3, R_3$ . Por cada  $i$ ,  $1 \leq L_i \leq R_i \leq 10^9$ .

### Salida

Imprime un número entero que represente la probabilidad de que Alice gane módulo  $10^9 + 7$ .

*Se puede demostrar que la respuesta se puede expresar como una fracción irreducible  $\frac{p}{q}$ , donde  $p$  y  $q$  son enteros y  $q \not\equiv 0 \pmod{10^9 + 7}$ , es decir, nos interesa el entero  $p \times q^{-1} \pmod{10^9 + 7}$ .*

<b>Ejemplo de entrada 1</b> 3 3 4 4 5 5	<b>Ejemplo de salida 1</b> 1
<b>Ejemplo de entrada 2</b> 4 4 8 8 12 12	<b>Ejemplo de salida 2</b> 0
<b>Ejemplo de entrada 3</b> 1 10 1 10 1 10	<b>Ejemplo de salida 3</b> 580000005
<b>Ejemplo de entrada 4</b> 5 15 2 9 35 42	<b>Ejemplo de salida 4</b> 1

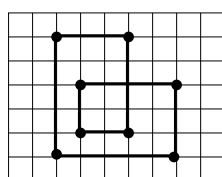
## Problema C

### Cortes con Laser

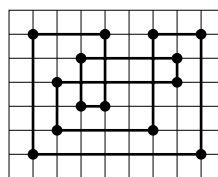
Una máquina laser para cortar hojas de madera tiene una cabeza laser que se puede mover en solo dos direcciones: horizontal y vertical. Has sido contratado para ser parte del equipo de pruebas de esta máquina.

Una de las pruebas consiste en programar la máquina para que realice una secuencia no vacía de cortes que inician y terminan en el mismo punto. Cada corte en la secuencia, excepto el primero, inicia en el punto en el que el corte previo finalizó. Ninguno de los cortes toca los bordes de la hoja que será cortada.

Las figuras (a) y (b) a continuación, muestran dos ejemplos de secuencias de corte con respectivamente 8 y 14 cortes.



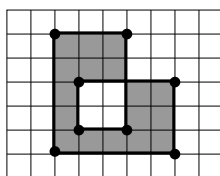
(a)



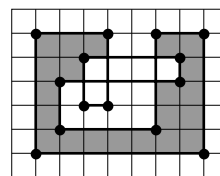
(b)

Tu jefe te ha pedido que determines el area de la pieza más grande producida por la secuencia de cortes, descartando la pieza que contiene a los bordes de la hoja de corte. Es decir, considerando solo las piezas contenidas en el polígono formado por las líneas de corte.

Las figuras (c) y (d) a continuación muestran respectivamente las piezas más grandes producidas por las secuencias de corte de las figuras (a) y (b).

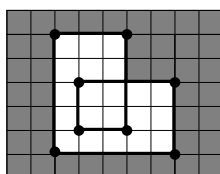


(c)

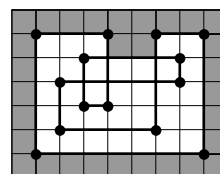


(d)

Para ilustrar, las figuras (e) y (f) a continuación muestran la pieza descartada (que contiene los bordes de la hoja de madera) de las secuencias de corte de las figuras (a) y (b) respectivamente.



(e)



(f)

## Entrada

La primera línea contiene un número entero  $N$ , el número de cortes en la secuencia ( $4 \leq N \leq 10^4$ ). La segunda línea contiene dos enteros  $X_0$  y  $Y_0$ , la posición inicial del cabezal láser en la secuencia de cortes ( $1 \leq X_0 \leq 10^3$  y  $1 \leq Y_0 \leq 10^3$ ). Cada una de las siguientes  $N$  líneas contiene dos enteros  $X_i$  y  $Y_i$ , la posición final del corte  $i$  ( $1 \leq X_i \leq 10^3$  y  $1 \leq Y_i \leq 10^3$ , con  $1 \leq i \leq N$ , y  $(X_N, Y_N) = (X_0, Y_0)$ ). Todas las posiciones dadas en la entrada son distintas, excepto para la primera y la última posición.

## Salida

Su programa debe imprimir una sola línea, que contenga un solo número entero, el área de la pieza más grande producida por la secuencia de cortes.

<b>Ejemplo de entrada 1</b> 8 2 1 7 1 7 4 3 4 3 2 5 2 5 6 2 6 2 1	<b>Ejemplo de salida 1</b> 17
<b>Ejemplo de entrada 2</b> 14 1 1 8 1 8 6 6 6 6 2 2 2 2 4 7 4 7 5 3 5 3 3 4 3 4 6 1 6 1 1	<b>Ejemplo de salida 2</b> 21

## Problema D

# Desplazamiento de Partículas

Un cuadrado tiene sus vértices en las coordenadas  $(0, 0)$ ,  $(0, 2^N)$ ,  $(2^N, 2^N)$ ,  $(2^N, 0)$ . Cada vértice tiene un atractor. Una partícula se coloca inicialmente en la posición  $(2^{N-1}, 2^{N-1})$ . Cada atractor se puede activar individualmente, cualquier número de veces. Cuando se activa un atractor en la posición  $(i, j)$ , si una partícula está en la posición  $(p, q)$ , se moverá al punto medio entre  $(i, j)$  y  $(p, q)$ .

Dado  $N$  y un punto  $(x, y)$ , calcula el número mínimo de veces que tienes que activar los atractores para que la partícula acabe en la posición  $(x, y)$ .

### Entrada

La entrada consiste de una sola línea que contiene tres enteros  $N$ ,  $x$  y  $y$ , tales que  $1 \leq N \leq 20$  y  $0 < x, y < 2^N$ .

### Salida

Imprima una sola línea, que contenga la menor cantidad de veces que tiene que activar los atractores.

<b>Ejemplo de entrada 1</b> 1 1 1	<b>Ejemplo de salida 1</b> 0
<b>Ejemplo de entrada 2</b> 4 12 4	<b>Ejemplo de salida 2</b> 1
<b>Ejemplo de entrada 3</b> 4 3 1	<b>Ejemplo de salida 3</b> 3

## Problema E

# Eliminando Globos

Una enorme cantidad de globos flotan en el Salón de Convenciones al finalizar la Ceremonia de Premiación del Concurso Subregional del ICPC. El gerente del Salón de Convenciones está enojado porque mañana será el anfitrión de otro evento y los globos deben ser removidos. Afortunadamente, este año Carlinhos vino preparado con su arco y flechas para reventar los globos.

Además, por suerte, debido al flujo de aire acondicionado, los globos están todos en el mismo plano vertical (es decir, un plano paralelo a una pared), aunque en distintas alturas y posiciones.

Carlinhos disparará desde el lado izquierdo del Salón de Convenciones, a una altura elegida, apuntando a la dirección del lado derecho del Salón de Convenciones. Cada flecha se mueve de izquierda a derecha, a la altura a la que se disparó, en el mismo plano vertical de los globos. Cuando una flecha toca un globo, el globo revienta y la flecha continúa su movimiento hacia la derecha a su altura disminuida en 1. En otras palabras, si la flecha estaba en la altura  $H$ , después de reventar un globo continuará en la altura  $H - 1$ .

Carlinhos quiere reventar todos los globos disparando la menor cantidad de flechas posible. ¿Puedes ayudarlo?

### Entrada

La primera línea de entrada contiene un entero  $N$ , el número de globos ( $1 \leq N \leq 5 \times 10^5$ ). Como todos los globos están en el mismo plano vertical, definamos que la *altura* de un globo se da en relación con el eje  $y$  y la *posición* de un globo se da en relación con el eje  $x$ . Los globos están numerados del 1 al  $N$ . Los números en los globos indican sus posiciones, desde el extremo izquierdo (globo número 1) hasta el extremo derecho (globo número  $N$ ), independientemente de sus alturas. La posición del globo número  $i$  es diferente de la posición del globo número  $i + 1$ , para todo  $i$ . La segunda línea contiene  $N$  enteros  $H_i$ , donde  $H_i$  indica la altura del globo número  $i$  ( $1 \leq H_i \leq 10^6$  para  $1 \leq i \leq N$ ).

### Salida

Su programa debe imprimir una sola línea, que contenga un solo número entero, el número mínimo de flechas que Carlinhos necesita disparar para reventar todos los globos.

<b>Ejemplo de entrada 1</b> 5 3 2 1 5 4	<b>Ejemplo de salida 1</b> 2
<b>Ejemplo de entrada 2</b> 4 1 2 3 4	<b>Ejemplo de salida 2</b> 4
<b>Ejemplo de entrada 3</b> 6 5 3 2 4 6 1	<b>Ejemplo de salida 3</b> 3

## Problema F

# Fanático del Ahorcado Multidimensional

El Juego del Ahorcado Multidimensional tiene reglas muy peculiares. En cierto modo, es como si estuvieras jugando varios juegos del tradicional juego del Ahorcado al mismo tiempo, con la diferencia de que las palabras no tienen por qué existir en el diccionario. Si nunca has jugado el juego del Ahorcado, no te preocupes: toda la información que necesitas está a continuación.

En la versión multidimensional del juego, hay varias palabras en un tablero, inicialmente desconocidas, todas de la misma longitud. En cada turno del juego, descubres algunos caracteres de ciertas posiciones de las palabras (el cómo estos caracteres fueron descubiertos no es importante para este problema). En un cierto momento del juego, cuando sólo queda un caracter desconocido en cada palabra del tablero, el juego entra en la fase de *todo o nada*. En este punto, se debe elegir una palabra que maximice el número de compatibilidades con las palabras del tablero. Para una palabra  $P$  elegida, decimos que es compatible con una palabra  $T$  en el tablero si todas las letras conocidas en  $T$  aparecen exactamente en las mismas posiciones en  $P$ .

Dada la información conocida sobre las palabras en el tablero, debe determinar qué palabra elegir en la fase de *todo o nada* que maximice el número de compatibilidades. Si hay más de una solución, escriba la menor lexicográficamente. Decimos que una palabra  $P$  es lexicográficamente más pequeña que una palabra  $Q$  si  $P_i < Q_i$  donde  $P_i$  es el  $i$ -ésimo caracter de  $P$ ,  $Q_i$  es el  $i$ -ésimo caracter de  $Q$  y  $i$  es el índice más pequeño tal que  $P_i \neq Q_i$ .

### Entrada

La primera línea de la entrada contiene dos enteros  $N$  y  $C$  que satisfacen  $1 \leq N \leq 10^4$  y  $1 \leq C \leq 12$ , indicando el número de palabras en el tablero y la longitud de las palabras que contiene. Cada una de las siguientes  $N$  líneas contiene una palabra de longitud  $C$  compuesta únicamente de caracteres de la “a” a la “z” excepto por una de sus posiciones, que contendrá un caracter “\*” , que indica que el caracter en esa posición aún se desconoce.

### Salida

Imprima una sola línea, que contenga, en orden, una palabra  $T$ , de longitud  $C$ , y un entero  $M$ , tal que  $M$  es el mayor número de compatibilidades que una palabra puede tener con las palabras de entrada y  $T$  es lexicográficamente la más pequeña entre las palabras con compatibilidad  $M$ .

<b>Ejemplo de entrada 1</b> 5 4 rat* ru*d rot* r*ta r*ta	<b>Ejemplo de salida 1</b> rata 3
<b>Ejemplo de entrada 2</b> 5 4 bon* fon* n*no *eto *ano	<b>Ejemplo de salida 2</b> nano 2

## Problema G

# Geometría Triangular

Todo polígono se puede construir uniendo triángulos. En particular, podemos hacer esto iterativamente: comenzamos con un triángulo, agregamos un segundo triángulo que identifica uno de sus lados con uno de los lados del triángulo inicial, añadimos un tercer triángulo identificando uno de sus lados a uno de los lados libres de alguno de los triángulos originales, y así sucesivamente. Solo consideraremos polígonos que se pueden construir de esta manera, donde cada triángulo agregado toca (y se identifica con) exactamente un lado de un triángulo colocado previamente.

Dado un polígono  $P$ , sea  $T$  el conjunto de triángulos utilizados para formarlo. Los lados de cada triángulo son segmentos de recta. Sea  $L$  el conjunto de segmentos que son lados de algún triángulo en  $T$ . Tenga en cuenta que cada elemento de  $L$  es un lado de uno o dos elementos de  $T$ .

Una vez que tenemos un polígono posicionado en el plano, en algunos casos podemos eliminar algunos de los triángulos que lo componen, sin cambiar el conjunto  $L$ . Queremos eliminar triángulos de tal manera que se mantenga el conjunto  $L$  sin cambios y el área total de los triángulos restantes sea mínima. De manera equivalente, queremos seleccionar un subconjunto  $S$  de triángulos de  $T$  tal que:

1. Cada elemento de  $L$  es el lado de al menos un triángulo en  $S$ ; y
2. La suma de las áreas de los elementos de  $S$  es lo más pequeña posible.

### Entrada

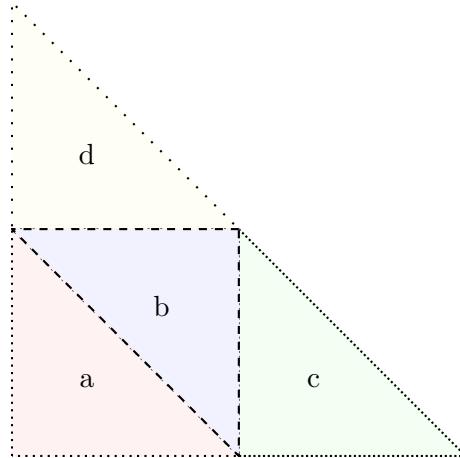
La primera línea de la entrada contiene un entero  $N$ ,  $1 \leq N \leq 10^5$  correspondiente a el número de triángulos en la triangulación de  $P$ . Cada una de las siguientes líneas  $N$  contiene 6 números,  $x_1, y_1, x_2, y_2, x_3$  y  $y_3$ , indicando la existencia de un triángulo con coordenadas  $(x_1, y_1)$ ,  $(x_2, y_2)$  y  $(x_3, y_3)$ . Los triángulos se dan en orden arbitrario. Todas las coordenadas serán números enteros con valor absoluto como máximo  $10^6$ .

### Salida

Imprima el área mínima posible, respetando las condiciones del problema, con exactamente un decimal.

<b>Ejemplo de entrada 1</b> 4 0 0 0 10 10 0 10 10 0 10 10 0 10 10 0 10 0 20 10 10 20 0 10 0	<b>Ejemplo de salida 1</b> 150.0
<b>Ejemplo de entrada 2</b> 3 0 0 0 10 10 0 10 10 0 10 10 0 10 10 20 0 10 0	<b>Ejemplo de salida 2</b> 150.0
<b>Ejemplo de entrada 3</b> 1 0 0 1 0 0 1	<b>Ejemplo de salida 3</b> 0.5





En la figura de arriba, las triangulaciones  $T_1 = \{a, b, c, d\}$  y  $T_2 = \{a, b, c\}$  representan, respectivamente, el primer y segundo ejemplo. Note como  $S_1 = \{a, c, d\}$  es un subconjunto válido en el primer caso. Aunque el triángulo  $b$  no se toma, todos sus lados están presentes en los triángulos seleccionados.

## Problema H

# Habilitando el movimiento

El presidente de Nlogonia decidió, por decreto, que todas las calles de Nlogonia deberían ser de un solo sentido. Debido a la falta de conocimientos en ciencia elemental, no hubo una planeación adecuada de los cambios. Después de que el nuevo sistema fue implementado, las personas no podían llegar al trabajo, o no podías regresar a sus casas desde su trabajo, por ejemplo. Como resultado, hubo caos y disturbios en muchas ciudades.

El presidente fue destituido y la nueva administración del país contrató a un equipo de científicos para resolver el problema. A su vez, el comité lo contrató a usted, un experto en complejidad de algoritmos, para ayudarlos con el cálculo eficiente de la solución.

Entonces, para cada ciudad, se te dan los *puntos de referencia* de la ciudad y las calles de un solo sentido, cada una de las cuales conecta dos puntos de referencia. Tu tarea es determinar la cantidad mínima de puentes de un solo sentido que se deben construir para tener una conectividad total en la ciudad. Cada puente también debe conectar dos puntos de referencia.

### Entrada

La primera línea de entrada contiene dos enteros,  $N$  y  $M$  ( $1 \leq N \leq 10^4, 1 \leq M \leq 10^6$ ), donde  $N$  es el número de puntos de referencia y  $M$  es el número de calles. cada una de las siguientes  $M$  líneas contiene dos enteros,  $R$  y  $S$ ,  $1 \leq R, S \leq N$ ,  $R \neq S$ , que corresponde a una calle que conecta  $R$  con  $S$ , por lo que cada vehículo en esa calle debe alejarse de  $R$ , hacia  $S$ .

### Salida

Su programa debe imprimir una sola línea que contenga el número mínimo de puentes que son necesarios para hacer felices a los habitantes.

<b>Ejemplo de entrada 1</b> 7 7 1 2 2 3 3 1 6 1 6 4 4 5 7 6	<b>Ejemplo de salida 1</b> 2
<b>Ejemplo de entrada 2</b> 7 7 2 1 3 2 1 3 1 6 4 6 5 4 6 7	<b>Ejemplo de salida 2</b> 2

<b>Ejemplo de entrada 3</b> 2 1 1 2	<b>Ejemplo de salida 3</b> 1
<b>Ejemplo de entrada 4</b> 3 3 1 2 2 3 3 1	<b>Ejemplo de salida 4</b> 0
<b>Ejemplo de entrada 5</b> 2 0	<b>Ejemplo de salida 5</b> 2
<b>Ejemplo de entrada 6</b> 6 4 1 2 1 3 4 6 5 6	<b>Ejemplo de salida 6</b> 3

## Problema I

# Información Interceptada

Spies Breaching Computers (SBC) es una agencia de espías digitales que está desarrollando un nuevo dispositivo para interceptar información usando ondas electromagnéticas, el cual, permite espiar incluso sin tener contacto físico con el objetivo.

Este dispositivo intenta obtener la información un byte a la vez, esto es, una secuencia de 8 bits donde cada uno de ellos, naturalmente, puede tener un valor de 0 o 1. En ciertas situaciones, debido a interferencia por otros dispositivos, la lectura no puede ser completada satisfactoriamente. En este caso, el dispositivo regresa un valor de 9 para el bit correspondiente, informando que la lectura no pudo ser realizada.

Para automatizar el reconocimiento de la información que el dispositivo obtiene, se ha realizado una solicitud para que se desarrolle un programa, el cual, dada la información leída por el dispositivo, informe si todos los bits fueron leídos de manera exitosa o no. Tu tarea es escribir este programa.

### Entrada

La entrada consiste de una sola línea, que contiene 8 enteros  $N_1, N_2, N_3, N_4, N_5, N_6, N_7$  y  $N_8$ , indicando los valores leídos por el dispositivo ( $N_i$  es 0, 1 o 9 para  $1 \leq i \leq 8$ ).

### Salida

Imprima una sola línea que contenga la letra mayúscula “S” si todos los bits fueron leídos satisfactoriamente; de lo contrario, imprima una sola línea que contenga la letra mayúscula “F”, indicando una falla.

<b>Ejemplo de entrada 1</b> 0 0 1 1 0 1 0 1	<b>Ejemplo de salida 1</b> S
<b>Ejemplo de entrada 2</b> 0 0 1 9 0 1 0 1	<b>Ejemplo de salida 2</b> F

## Problema J

# Jugando 23

Veintitrés es un juego simple de cartas que juegan los niños. Como su nombre lo indica, es una variación del juego veintiuno (*blackjack*) en inglés, el cual es uno de los juegos más jugados en casinos y sitios de juegos.

En veintitrés gana el jugador que tiene más puntos siempre y cuando no supere los 23. Si un jugador tiene más de 23 puntos decimos que el jugador *reventó*.

El juego utiliza una baraja de 52 cartas, con cuatro palos, cada palo con 13 cartas (as, 2, 3, 4, 5, 6, 7, 8, 9, 10, jota, reina y rey). Los palos de cartas no son relevantes. Las cartas con imagen (jota, reina y rey) valen diez puntos, las cartas con números valen su número en puntos (por ejemplo, la carta 4 vale cuatro puntos) y el as vale un punto.

Las reglas del juego son simples: en cada juego, inicialmente se barajan las cartas del mazo, las cartas se colocan en una pila y cada jugador recibe dos cartas de la pila. Todas las cartas se reparten boca arriba (todos los jugadores ven las cartas de todos los jugadores). El siguiente paso, llamado *ronda*, se repite mientras haya jugadores activos: se toma una carta de la pila, boca arriba. Esta carta, llamada *carta común*, se aplica a todos los jugadores. Si un jugador revienta, deja el juego. El ganador será aquel jugador que en una determinado ronda sume 23 (sumando sus dos cartas iniciales más las cartas comunes), o si el jugador es el único jugador activo al final de la ronda. Tenga en cuenta que puede haber más de un ganador (cuyas cartas suman 23) y que puede no haber ganador en un juego.

Juan y María están jugando veintitrés. Los dos son los únicos jugadores, ninguno de ellos reventó y ninguno de ellos tiene 23 puntos. Además, la puntuación de los jugadores es tal que la siguiente carta común puede hacer que el juego termine.

Dadas las cartas iniciales de Juan y María y las cartas comunes, determine ¿Cuál es el valor de la carta de menor valor que se debe tomar de la pila en la siguiente ronda para que María gane el partido?

### Entrada

La primera línea de entrada contiene un solo número entero  $N$  ( $1 \leq N \leq 8$ ), el número de rondas jugadas hasta ese momento. Cada carta está descrita por un entero  $I$  ( $1 \leq I \leq 13$ ). Tenga en cuenta que las cartas con imagen (jota, reina y rey) están representadas en la entrada por los valores 11, 12 y 13, que no es cuántos puntos valen. La segunda línea contiene dos enteros, que describen las dos cartas iniciales de Juan. La tercer línea contiene dos números enteros, que describen las dos cartas iniciales de María. La cuarta y última línea contiene  $N$  enteros, que describen cada una de las cartas comunes, en el orden en que fueron tomadas de la pila.

### Salida

Su programa debe imprimir una sola línea, que contenga un solo número entero, el valor de la carta de menor valor que debe tomarse de la pila en la siguiente ronda para que María gane el juego, o -1 si María no puede ganar el juego en la siguiente ronda.

Ejemplo de entrada 1	Ejemplo de salida 1
1 10 5 9 10 1	3

<b>Ejemplo de entrada 2</b> 1 10 5 8 7 2	<b>Ejemplo de salida 2</b> 6
<b>Ejemplo de entrada 3</b> 1 9 10 10 5 1	<b>Ejemplo de salida 3</b> 4
<b>Ejemplo de entrada 4</b> 2 8 4 4 1 4 4	<b>Ejemplo de salida 4</b> 5
<b>Ejemplo de entrada 5</b> 8 2 1 1 1 1 2 2 2 3 3 3 3	<b>Ejemplo de salida 5</b> -1

## Problema K

# Kalel, la rana saltarina

Kalel es una rana a la que le gusta saltar sobre piedras.

Hay  $N$  piedras en una fila, numeradas del 1 al  $N$  de izquierda a derecha. Kalel comienza en la piedra 1 y quiere llegar a la piedra  $N$ .

En cada movimiento, Kalel puede elegir entre  $M$  tipos de salto. El  $j$ -ésimo salto le permite saltar de la piedra  $x$  a la piedra  $x + d_j$  y cuesta  $p_j$  puntos de energía. Puede suceder que  $p_j$  sea igual a 0 para algunos  $j$ . Puedes suponer que Kalel nunca se queda sin energía.

Dados  $N$  y  $K$ , calcula de cuántas maneras Kalel puede alcanzar la piedra  $N$  gastando como máximo  $K$  puntos de energía en total. Dos formas se consideran diferentes si la secuencia de opciones de salto es diferente. Como este número puede llegar a ser muy grande, solo nos interesa su resto módulo  $10^9$  (mil millones).

### Entrada

La primera línea contiene tres enteros,  $N$ ,  $M$  y  $K$  ( $1 \leq N \leq 10^9$ ,  $1 \leq M \leq 10^5$ ,  $0 \leq K \leq 400$ ). Las siguientes  $M$  líneas contienen dos enteros cada una, los números  $d_j$  y  $p_j$  ( $1 \leq d_j \leq 10$ ,  $0 \leq p_j \leq K$ ).

### Salida

Imprima una sola línea, que contenga de cuántas maneras diferentes Kalel puede llegar a la roca  $N$  gastando un máximo de  $K$  puntos de energía, módulo  $10^9$  (mil millones).

<b>Ejemplo de entrada 1</b> 5 3 10 1 3 2 0 3 1	<b>Ejemplo de salida 1</b> 6
<b>Ejemplo de entrada 2</b> 100000 3 10 1 9 2 0 7 3	<b>Ejemplo de salida 2</b> 85449877

## Problema L

# Listando caminos tediosos

Un árbol es un grafo conectado (existe un camino entre cualquier par de vértices), no dirigido (las aristas del grafo no tienen dirección) y acíclico (no tiene ciclos).

Un árbol colorido es un árbol en el que cada uno de sus vértices tiene un color específico.

Un camino tedioso es un camino en el árbol tal que tanto el vértice inicial como el final tienen el mismo color, y no hay vértice o aristas que aparezcan más de una vez en el camino. Tenga en cuenta que el color de los vértices intermedios, si los hay, son irrelevantes.

Dado un árbol colorido, con  $N$  vértices, tu tarea es calcular, para cada una de las aristas, el número de caminos tediosos que pasan por esa arista.

### Entrada

La primera línea de entrada contiene el número de vértices  $N$  ( $1 \leq N \leq 10^5$ ). La segunda línea contiene  $N$  enteros  $C_1, \dots, C_N$ , donde  $C_i$  ( $1 \leq C_i \leq N$ ) representa el color del vértice  $i$ . Las siguientes  $N - 1$  líneas contienen dos enteros cada una,  $u$  y  $v$ , que representan una arista ( $1 \leq u, v \leq N$  y  $u \neq v$ ). Se garantiza que el grafo dado es un árbol.

### Salida

Imprime  $N - 1$  enteros, que representan el número de caminos tediosos que pasan por cada arista, siguiendo el mismo orden de las aristas que se dan en la entrada.

<b>Ejemplo de entrada 1</b> 6 1 1 1 2 2 1 1 2 2 3 4 6 2 4 1 5	<b>Ejemplo de salida 1</b> 4 3 3 4 1
<b>Ejemplo de entrada 2</b> 12 1 2 3 1 2 2 1 3 2 3 1 2 1 2 2 3 2 4 4 5 4 6 1 7 7 8 7 9 9 10 6 11 6 12	<b>Ejemplo de salida 2</b> 10 2 10 4 9 9 2 6 2 3 4

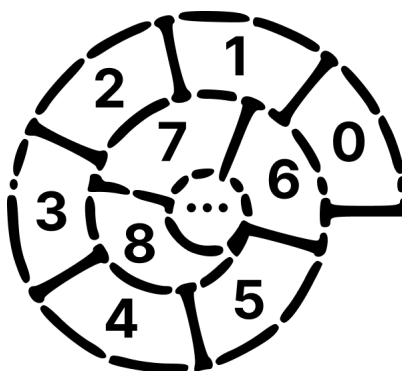


## Problema M

# Maratón de Rayuela

8 de octubre de 2022. Esta es la fecha del evento más esperado del año por los estudiantes de informática de todo el país. No, no estamos hablando del ICPC.

¡Estamos hablando, por supuesto, de Rayuela! Para aquellos que no estén familiarizados, Rayuela es una competencia anual que tradicionalmente se lleva a cabo como un evento paralelo del ICPC. Transmitida en vivo para espectadores de todos los continentes y para practicantes de los lenguajes de programación más esotéricos, esta variante exótica del popular juego infantil tiene lugar en una cancha infinita en forma de espiral, subdividida en áreas numeradas secuencialmente comenzando en cero, como se muestra a continuación.



Este año, Rayuela ha atraído un número récord de  $N$  participantes, numerados secuencialmente de 1 a  $N$ . Se sabe que el  $i$ -ésimo participante comienza en el área numerada  $A_i$ .

Rayuela consta de  $Q$  rondas. Durante la  $q$ -ésima ronda, Carlão, amado organizador de Rayuela desde hace más tiempo del que nadie puede recordar, comunicará dos números enteros a los participantes:  $c_q$  y  $d_q$ . Esta es una orden para todos los participantes con un número de identificación  $i$  tal que  $i$  y  $c_q$  comparten un factor entero común mayor que 1 para retroceder  $d_q$  posiciones en la cancha Rayuela, una por una, sin retroceder nunca más allá que la posición 0. (Cualquier participante que eventualmente regrese a la posición 0 debe permanecer allí indefinidamente, ignorando cualquier otra orden de retroceso, para no abandonar la cancha).

Bajo el supuesto de que los participantes han ejecutado las instrucciones perfectamente (nunca querrían decepcionar a Carlão), tu tarea es determinar, para cada participante, el número de la ronda en la que él o ella llegará a la posición 0 (o indicar de otra manera que esto nunca suceda).

### Entrada

La primera línea contiene los enteros  $N$  y  $Q$  ( $1 \leq N, Q \leq 10^5$ ). La segunda línea contiene  $N$  enteros,  $A_1, A_2, \dots, A_N$  ( $1 \leq A_i \leq 10^9$ ). Cada una de las siguientes  $Q$  líneas contiene dos números enteros,  $c_q$  y  $d_q$  ( $1 \leq c_q \leq 10^5$ ,  $1 \leq d_q \leq 10^9$ ).

### Salida

Imprima  $N$  líneas. La  $i$ -ésima línea debe contener un solo número entero, que indica el número de la ronda cuando el  $i$ -ésimo participante llegará a la posición 0 (o el valor  $-1$ , si eso nunca sucede).

<b>Ejemplo de entrada 1</b> 7 6 10 20 30 40 50 60 70 2 25 3 36 100 42 5 10 7 70 1 1000	<b>Ejemplo de salida 1</b> -1 1 2 3 4 2 5
<b>Ejemplo de entrada 2</b> 6 4 100 100 100 100 100 100 2 50 3 50 5 99 5 1	<b>Ejemplo de salida 2</b> -1 -1 -1 -1 4 2

## Problema N

# Números al otro lado

Acabas de ganar una baraja de cartas con  $N$  cartas. Cada una de estas cartas tiene dos números escritos: uno al frente y otro al reverso.

Tu amigo te ha retado a un juego. Barajó las cartas y las puso en una mesa. Las cartas se colocan en una línea, lado a lado, con el frente hacia arriba.

De izquierda a derecha, sabes que el número escrito en el frente de la  $i$ -ésima tarjeta es  $A_i$ , y que el número escrito en el reverso de la  $i$ -ésima tarjeta es  $B_i$ .

El juego se divide en dos partes.

En la primera parte debes elegir  $K$  cartas de la baraja. Para elegir una carta, debe elegir la primera carta a la izquierda, o la primera carta a la derecha de la mesa, y recogerla.

Después de eso, debes elegir  $L$  de las cartas que recogiste y voltearlas.

Su puntaje será igual a la suma de los números escritos en el frente de todas las  $K$  cartas que escogiste, más la suma de los números escrito al reverso de las  $L$  cartas que volteó.

¿La meta? ¡Lograr la puntuación más alta posible, por supuesto!

### Entrada

La primera línea contiene un entero  $N$  ( $1 \leq N \leq 10^5$ ). La segunda línea contiene  $N$  enteros  $A_1, A_2, \dots, A_N$ , ( $1 \leq A_i \leq 10^9$ ). La tercera línea contiene  $N$  enteros  $B_1, B_2, \dots, B_N$ , ( $1 \leq B_i \leq 10^9$ ). La cuarta línea contiene dos enteros  $K$  y  $L$  ( $1 \leq L \leq K \leq N$ ).

### Salida

Imprima una sola línea que contenga un entero, representando la puntuación más alta posible.

<b>Ejemplo de entrada 1</b> 5 9 7 2 2 9 5 2 2 3 1 2 1	<b>Ejemplo de salida 1</b> 23
<b>Ejemplo de entrada 2</b> 5 9 7 2 2 9 5 9 2 3 1 2 1	<b>Ejemplo de salida 2</b> 25