Maratona SBC de Programação 2024

This problem set is used in simultaneous contests:
Competencia Boliviana de Programación
The 2024 ICPC Gran Premio de Centroamerica
The 2024 ICPC Gran Premio de Mexico

August 31st, 2024

Problems book

General Information

This problem set contains 12 problems; pages are numbered from 1 to 16, without considering this page. Please, verify your book is complete.

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

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

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

Promo:

Sociedade Brasileira de Computação

v1.0

Problem A
Attention to the Meeting

Vinicius is at a board meeting of the "Instituto de Consultoria de Palestras e Comentários" (ICPC) thinking that it would be great if the board members were more concise and kept their speeches within the time allotted for each director, so that the meeting could end before lunch. Unfortunately, perhaps due to the nature of the institution, everyone loves to talk.

Knowing that

- there are N directors who will speak at the meeting;
- each director will speak for the same amount of time;
- and that between two consecutive speeches there is a 1-minute interval,

determine the maximum length of each speech, in minutes, so that the meeting lasts no more than K minutes.

Input

The first line contains an integer N ($1 \leq N \leq 100$), the number of directors. The second line contains an integer K ($1 \leq K \leq 1000$ and $K \geq N$), the maximum meeting duration in minutes. For all input cases, each director's speech lasts at least 1 minute.

Output

Your program should output a single line, containing a single integer, indicating the length of each board member's speech, in minutes.

Input example 1	**Output example 1**
`7` `120`	`16`

Explanation of sample 1:

There are 7 directors and the maximum meeting length is 120 minutes. If each director speaks for 16 minutes, we have $16 \times 7 = 112$ minutes. Since there are six breaks between speeches, and each break lasts one minute, we have 118 minutes in total. Note that, in this case, two minutes of the meeting time are not used, and that, if the speeches were longer than 16 minutes, the total time would exceed the 120-minute limit.

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

Explanation of sample 2:

There is only one director and the meeting lasts 10 minutes. Therefore, the maximum speaking time of the director is 10 minutes.

Input example 3	**Output example 3**
`100` `1000`	`9`

Problem B
Bacon Number

Carlinhos loves movies, and recently he has been fascinated by the *Bacon Number*, which is defined as follows.

- The Bacon number of the actor Kevin Bacon is equal to 0;
- If the smallest Bacon number of an actor with whom X has appeared in the same movie is b, the bacon number of the actor X is $b + 1$.

That is, the Bacon number measures the shortest path between any actor and the actor Kevin Bacon, in which two actors are connected if they appeared together in the same movie.

Carlinhos is interested in a more general problem: given two actors, how to connect them through intermediate movies and actors? Given N movies, and, for each movie, which of the existing M actors acted in it. Carlinhos wants to answer Q queries: in the i-th of them, we want to compute some way to connect actor x_i with actor y_i. We must find some sequence $x_i = a_1, f_1, a_2, f_2, \ldots, f_{k-1}, a_k = y_i$, where $1 \leq a_j \leq N$ are actors and $1 \leq f_j \leq M$ are movies, and actor a_j acted in movies f_{j-1} and f_j, or indicate that no such sequence exists.

Input

In the first line of the input, two integers N $(1 \leq N \leq 100)$ and M $(1 \leq M \leq 10^6)$ are given, the number of movies and the number of actors.

N lines follow. In the i-th line, the first integer n_i $(1 \leq n_i \leq M)$ denotes the number of actors in movie i. Next, n_i numbers in ascending order separated by spaces: the indices, from 1 to M, of the actors who acted in movie i.

The next line, contains an integer Q $(1 \leq Q \leq 10^4)$: the number of queries.

The next Q lines describe the queries. In the i-th of them, read two numbers x_i, y_i $(1 \leq x_i \neq y_i \leq M)$, the actors we want to connect. It is guaranteed that the total number of actors in the movies is at most 10^6. That is, $\sum_i n_i \leq 10^6$.

Output

For each of the queries, if there is no sequence, print a line with -1. Otherwise, print two lines. In the first line, print the number of actors k_i $(2 \leq k_i \leq 10^6)$ in some way to connect x_i and y_i. In the second, print the sequence as described, with k_i actors and $k_i - 1$ movies, alternating. If there is more than one way to connect the actors, print any of them.

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

Problem C
Couple of BipBop

It's time for Bob and Charlie to go on a new couple-hyperfocus: BipBop trends. This social network specialized in short videos is going viral more than ever before. As a consequence, couples now measure how much they love each other in terms of how well they can dance together. In theory, the BipBop dancing style is simple and can be used to perform pretty much every song existent. Usually, it consists in a sequence of moves, one for each verse, represented by an integer number, as the moves are kinda generic, really.

Always late, the couple just got to a party, the song is already playing, but they still want to impress and show that they can dance BipBop even without knowing in what verse the song is currently at. Then, each of them starts dancing in a random verse and keep following the coreography until one of them reaches the end of the song or when they unmatch a move (they execute different moves).

There is no popular song that Bob and Charlie don't know how to dance, so given a song represented as a sequence of movements, one for each verse, calculate the expected number of verses they will be dancing in sync, if each of them initially thinks that the song is playing on a random verse with uniform probability.

Input

The first line of the input contains an integer N ($1 \leq N \leq 10^5$), the number of verses in the song. The second line contains N integers, $V_1, V_2, \ldots, V_N$ ($1 \leq V_i \leq N$), corresponding to the movement associated with each of the verses in the sequence.

Output

Output the expected number of verses (moves) the couple will dance in sync, if each one of them chose a verse uniformly at random to start the dance. Output the answer as a irreducible fraction P/Q, such that $\gcd(P, Q) = 1$. It can be proven that it is always possible to express the answer in this way.

Input example 1	Output example 1
2 1 1	5/4

Explanation of sample 1:

Note that there are 4 equally likely ways for the choreography to occur: both Bob and Charlie can start on the first or second verse, with probability 1/2 that each will start on each of the verses and therefore probability 1/4 for each of the combinations. If both start on the first verse, they will dance 2 verses in sync. In the other three possibilities, they will dance only one verse in sync. Thus, we have on average, $2 \times 1/4 + 1 \times 1/4 + 1 \times 1/4 + 1 \times 1/4 = 5/4$ verses in sync.

Input example 2	Output example 2
4 1 1 1 1	15/8

Input example 3	Output example 3
7 1 2 1 3 1 2 1	48/49

Problem D
Decrease the Boss Strength

Fulano, an avid gamer, has come across an epic challenge in the online game "Boss Challenge". The goal is to defeat a powerful boss, whose power is described by a set of ancient runes. These runes represent a giant binary number N, indicating the total strength of the enemy.

To defeat the boss, Fulano has M different spells at his disposal, and the goal is to reduce the total strength of the enemy to zero using these spells. The i-th spell is described with two integers a_i and b_i. When used, the i-th spell reduces the value of N by a_i units. This spell can be used as many times as the player wants, as long as two specific conditions are met:

- The value of a_i must be less than or equal to the current value of N.
- The current value of N must be divisible by 2^{b_i}. In other words, the spell i can only be used if the last b_i digits of N are zeros.

Fulano is fascinated by the game and wants to find out how many different ways he can combine the spells to reduce the binary number N to exactly zero and thus defeat the boss. Two combinations are considered different if the sequence in which the spells are used is different.

Since the number of possible combinations can be very large, the answer should be given modulo $10^9 + 7$.

Help Fulano find the answer!

Input

The first line contains a single integer N ($1 \leq N \leq 10^{18}$), representing the boss's power.

The second line contains a single integer M ($1 \leq M \leq 10^5$), denoting the number of spells available.

The next M lines contain the spell descriptions: the i-th of these lines contains two numbers a_i ($1 \leq a_i \leq 100$) and b_i ($0 \leq b_i \leq 60$).

Output

Print a single integer: the number of different sequences of spell uses (taken modulo $10^9 + 7$) that reduce the boss's power from N to 0.

Input example 1	Output example 1
6 2 1 0 2 1	8

Input example 2	Output example 2
9 5 1 0 1 1 4 3 1 1 8 0	92

Problem E

Enigma of the Jewelry Case

The princess of Nlogonia keeps her pearl collection in a square jewelry case made up of N columns, each column containing N small boxes. She places a different number of pearls in each box, and arranges the box so that in each column, from top to bottom, the boxes contain an increasing number of pearls and in each row, from left to right, the boxes also contain an increasing number of pearls.

The princess suspects that her little sister, who is very mischievous, is messing with her things in her games. In particular, the princess suspects that her jewelry case has been rotated 90 degrees clockwise, possibly multiple times.

Figure (a) below shows an example of the original arrangement of a 4×4 case. Figure (b) shows the case rotated clockwise, 90 degrees, once.

3	4	6	12
7	10	11	19
9	14	17	22
15	16	20	25

(a)

90°

15	9	7	3
16	14	10	4
20	17	11	6
25	22	19	12

(b)

Given the number of pearls in each box, write a program to determine the smallest number of 90-degree counterclockwise rotations that are necessary to return the jewelry case to its original state.

Input

The first line of the input contains an integer N, the number of rows and columns in the case ($2 \leq N \leq 50$). Each of the following N lines contains N integers $K_{i,j}$, the number of pearls in the box in row i and column j ($0 \leq K_{i,j} \leq 10^5$, for $1 \leq i \leq N$ and $1 \leq j \leq N$). In the input, the rows are given from top to bottom, and the columns are given from left to right.

Output

Your program should output a single line containing only one integer R (which can be 0, 1, 2, or 3), the smallest number of times the jewelry case must be rotated counterclockwise to return to its original state.

Input example 1	**Output example 1**

`4`
`15 9 7 3`
`16 14 10 4`
`20 17 11 6`
`25 22 19 12`
	`1`

Explanation of sample 1:

This example corresponds to the example in the statement. It is necessary to rotate the case counterclockwise once.

Input example 2	**Output example 2**

`3`
`300 250 150`
`280 200 140`
`240 190 130`
	`2`

Explanation of sample 2:

It is necessary to rotate the case counterclockwise twice.

Input example 3	**Output example 3**

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

Explanation of sample 3:

It is necessary to rotate the case counterclockwise three times.

Problem F
Fractions are better when continued

Little Charles was one of the best competitive programmers in the world. However, he never really liked programming. Now that he is retired, he can dedicate his studies to what he really loves: continued fractions.

To prepare for the upcoming Imensa Competição de Phrações Contínuas (ICPC), he needs to solve the following problem:

Define $p_0 = 1$ as the level 0 fraction. Then define:

$$p_1 = \frac{1}{1+1}$$

as the level 1 fraction, p_1. And also,

$$p_2 = \frac{1}{1+\frac{1}{1+1}}$$

as the level 2 fraction, p_2, and so on.

Given an integer value N, help Charles determine the value of the numerator of the fraction p_N.

Input

The first and only line contains an integer N ($1 \le N \le 40$).

Output

The value p_N can be written as a fraction of the form $\frac{a}{b}$, where a and b are coprime. Print a line containing the value of a.

Input example 1	**Output example 1**
`2`	`2`

Input example 2	**Output example 2**
`10`	`89`

Problem G
Geography of Rivers

When studying the geography of the world's rivers, you may ask yourself: when two rivers join together, who chooses the name of the river that results from this junction? In fact, the answer is simple: when two rivers join together, the name of the river that had the largest volume of water becomes the name. Given that all rivers eventually join together and flow into the sea, an interesting problem is to calculate, given the name of each source, the name of the final river that flows into the sea.

Formally, N river sources are given. For each source, you have a quantity of liters of water l_i that originates from it. Furthermore, pairs of rivers meet (like a binary tree), until they all join and flow into the sea. When two rivers meet, the quantity of liters of water is added together, and the name of the river becomes the name of the river that had more water, or, in case of a tie, the one with the lowest index. The initial name of each source is its index.

What you want to know is the name of the river that eventually flows into the sea. However, it's rainy season! You need to process Q operations. In each of them, a rain occurred that caused q_i liters **more** of water to be produced in the source n_i (and this will be maintained for future operations). After each operation, calculate the name of the river that flows into the sea.

Input

The first line contains an integer N ($1 \leq N \leq 10^5$): the number of river sources.

The second line contains N integers l_i ($1 \leq l_i \leq 10^9$): the number of liters of water that originate in source i.

The following $N-1$ lines describe how the rivers join together. In the i-th of them, two integers a_i, b_i ($1 \leq a_i, b_i < N+i$) indicate that the rivers a_i and b_i join together to form the river $N+i$ (whose volume of water will be the sum of the volumes of a_i and b_i, and whose name will be the name of the one with the largest volume of water). It is guaranteed that the values a_i and b_i are valid, that is, $a_i \neq b_i$ and neither of them has been previously joined in the input.

The next line contains an integer Q ($1 \leq Q \leq 10^5$), the number of operations.

Then Q lines with the operations follow: the i-th line contains two integers n_i and q_i ($1 \leq n_i \leq N$ and $1 \leq q_i \leq 10^9$), meaning that the source n_i now sources q_i liters **more** of water.

Output

Print, on the first line, the name of the river that initially flows into the sea. Then print Q lines: after each operation, the name of the river that flows into the sea.

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

Problem H
Harmonics with Interference

The transmission of messages by electromagnetic means presents several challenges, such as interference from other natural or artificial signals that can corrupt a transmission.

A common strategy is to send additional information that allows a received message to be validated. Some more robust protocols even allow for the correction of some errors in the sent message.

Arthur and Bruna are testing a new transmission protocol on a device they have developed. A message M, which is a sequence of bits, is sent from Arthur to Bruna, along with a control sequence N, also represented as a sequence of bits. By composing the message M and choosing the bits from N, Arthur ensures that the integer encoded by M is divisible by the integer represented by N.

For each bit received by Bruna, if the bit was transmitted without problems, the value `0` or `1` will be stored in the receiving device. If there was any interference, the symbol `*` is stored in place of the bit. The result of the transmission will be stored in the pair (M', N').

After the communication, if the message was sent successfully, Bruna can decode the original message M (since $M = M'$). If there was a problem, due to the way the protocol works, it may still be possible to decode the message. If many bits were lost, Bruna simply discards the message. But for transmissions where at most 16 bits of the original pair (M, N) were lost, Bruna would like to try to recover the message, avoiding retransmissions. She needs your help to recover one of the possible messages encoded by the received pair (M', N').

For example, suppose Bruna received M'=`111*` and N'=`1*`. Two transmissions could have been made:

1. M=`1111` with N=`11`. In this case, the numbers 15 and 3 are represented by M and N, respectively.
2. M=`1110` with N=`10`. In this case, the numbers 14 and 2 are represented by M and N, respectively.

Your task is: given the representations of the information received, find a message M that could have been sent by Arthur. If more than one message exists, you can print any message that could have been transmitted by Arthur.

Input

The first line of input will contain a sequence of characters representing M', with $1 \leq |M'| \leq 500$. The second line of input will contain a sequence of characters representing N', with $1 \leq |N'| \leq 16$. All characters in N' and M' will be either `0`, `1`, or `*`. In total, there will never be more than 16 `*` characters in the input. It is guaranteed that N' always contains at least one bit 1.

Output

A single line should be printed, containing a message M, compatible with the information received by Bruna.

Input example 1	**Output example 1**
`111*` `1*`	`1111`

Explanation of sample 1:

This case corresponds to the example given in the statement.

Input example 2	**Output example 2**
`101**` `11`	`10101`

Explanation of sample 2:

In this case, the different ways of choosing the unknown bits would result in messages corresponding to the integers 20, 21, 22 and 23, and only 21, represented by `10101`, is divisible by 3.

Problem I
Ingredients that may Harm You

In Nlogonia, foods are identified by numbers. Prime numbers identify the basic ingredients, and the number that identifies each food is given by the product of the numbers associated with the ingredients that compose it, respecting multiplicities. For example, a food with the number 12 contains two units of the ingredient 2, and one unit of the ingredient 3, since $12 = 2 \cdot 2 \cdot 3$.

You live in Nlogonia, and you own a self-service restaurant, that is, where people assemble their own dishes with the food available in the restaurant. You are expecting to serve Q people in your restaurant today.

Each person has a set of allergies, which are identified by an integer in the same way: each prime number that divides the person's number indicates that he or she is allergic to the ingredient associated with that prime number.

Given the numbers associated with each food item in your restaurant, calculate, for each of the Q people, how many different dishes she can assemble so that there is no ingredient in the dish to which she is allergic.

Input

The first line of the input contains an integer N ($1 \leq N \leq 10^5$), the number of foods in your restaurant. The next line contains the numbers associated with each food V_i ($1 \leq V_i \leq 10^6$). The next line contains an integer Q ($1 \leq Q \leq 10^5$), the number of people who will eat at your restaurant. Q lines follow; the i-th of them contains a number X_i ($1 \leq X_i \leq 10^6$), the number representing the allergies of person i.

Output

For each of the Q people, print a number: the number of dishes that can be assembled with the restaurant's ingredients, so that none of the ingredients to which the person is allergic are present. Since the answer may be very large, print the remainder when dividing it by $10^9 + 7$.

Input example 1	Output example 1
6 1 2 3 4 5 6 4 1 2 4 6	64 8 8 4

Explanation of sample 1:

The first person has no allergies, so all 64 possible dishes are valid for her. On the other hand, the last person is allergic to foods that contain the ingredients associated with the prime numbers 2 and 3. Therefore, only 4 dishes are possible for her: the empty plate (without any food), the plate with only food 1 (which has no ingredients), the plate with food 5, and the plate with foods 1 and 5.

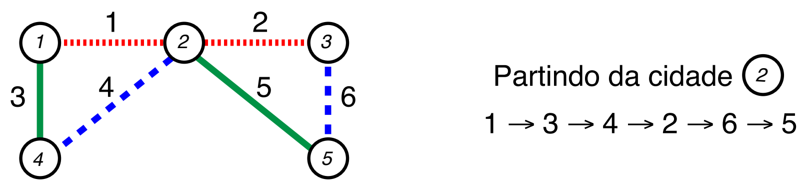
Problem J
Journey through Colors

In the land of Oz, roads are paved with colored stones. Each road connects exactly two cities, can be traveled in both directions, and is colored with stones of a single color.

Dorothy is visiting Oz for the first time and wants to take a tour of the country, meeting the following conditions:

- The tour must start and end in the same city.
- The tour must pass through each road in the country exactly once and cannot use two consecutive roads (i.e., one immediately after the other) that have the same color.
- The first and last roads of the tour must have different colors.

Figure (a) below illustrates an example with five cities and six roads. Figure (b) shows a possible tour that starts and ends in city 2 and satisfies the road color restrictions. In figure (b), the tour starts in city 2 and goes, in sequence, through roads 1 (red), 3 (green), 4 (blue), 2 (red), 6 (blue) and, finally, 5 (green).



(a) (b)

Help Dorothy find such a tour or, if it is not possible, indicate that it does not exist.

Input

The first line of the input contains three integers, N, M, and K, representing the number of cities ($2 \leq N \leq 1000$), the number of roads ($1 \leq M \leq 1000$), and the number of colors ($1 \leq K \leq 1000$), respectively. Cities are identified by integers from 1 to N, roads are identified by integers from 1 to M, and colors are identified by integers from 1 to K. Each of the following M lines describes a road and contains three integers I, J, and C, where I and J represent cities ($1 \leq I, J \leq N$, and $I \neq J$), and C indicates the color of road $1 \leq C \leq K$. The roads are given in the order of their identification, that is, the first road in the input is number 1, the second road is number 2, and so on.

Output

If there is no tour that satisfies the constraints, print a single integer -1. Otherwise, your program should output two lines describing a valid tour. The first line should contain the identifier of the starting city of the tour. The second line should contain M distinct integers, each identifying a road, in tour order. If there is more than one possible tour, print any one of them.

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

Explanation of sample 1:

This is the example from the statement. There are five cities, six roads and three colors (1 = red, 2 = green, 3 = blue). Note also that there are other possible tours, for example starting from city 1: $3 \to 4 \to 2 \to 6 \to 5 \to 1$.

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

Input example 3	**Output example 3**
6 6 3 1 2 1 2 3 2 3 1 3 4 5 1 5 6 2 6 4 3	-1

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

Input example 5	**Output example 5**
3 3 1 1 2 1 2 3 1 3 1 1	-1

Problem K
Karamell

Karamell, Caramel, Caramello or Caramelo. Different languages, but you know what I'm talking about. Alice and Bob are twins and they also love caramels! So, as a birthday present, they asked for caramels to all the guests at the party they are organizing.

The day of the party Alice and Bob received their presents: N bags of caramels. The i-th bag contained a_i caramels.

Alice and Bob don't want to open the bags right away, they decided to distribute the caramels in the following way: the bags will be considered in order and, at the i-th step, the a_i caramels from the i-th bag are given to the person who has the least caramels at that moment. In case of a tie, Alice gets the caramels (after all, "ladies first").

One thing they didn't like is that, depending on the order in which the bags are considered, the final amount of caramels that each person receives can be different. For example, if the bags were ordered in the quantities described by the sequence $[1, 2, 2, 3]$, Alice would end up with 3 and Bob would end up with 5 candies. On the other hand, if they were considered in the order $[1, 2, 3, 2]$, both would end up with 4.

You forgot to buy candies for the birthday children, but you decided to give them an even more interesting gift: a program that determines a way to order the bags so that Alice and Bob get the same amount of candies, if possible.

Input

The first line contains a single integer N ($1 \leq N \leq 100$), indicating the number of bags. The second line contains N integers $a_1, \ldots, a_N$ ($1 \leq a_i \leq 100$), where a_i indicates the number of candies in bag i.

Output

The output must be a single line. If it is impossible to find an order as requested, print -1. Otherwise, print N integers separated by spaces, indicating a valid ordering of the a_i values that guarantees that the candies will be divided equally among the siblings.

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

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

Input example 3	Output example 3
6 1 12 21 23 33 34	-1

Problem L
Lecographically Maximum

A list of N integers $a_1, \ldots, a_N$ is stored in the memory of an electronic device. This device has a very peculiar operation available: bit swapping between numbers. More precisely, given integers i, j and k, this operation swaps the k-th bit of the integer a_i with the k-th bit of the integer a_j (and vice-versa).

Very interesting phenomena can occur when performing this operation one or more times, such as obtaining numbers that did not even belong to the original list, or even numbers larger or smaller than all the original elements.

For this problem, we are interested in using the operation as many times as necessary to change the list of numbers so that the resulting list is the lexicographically maximum, that is, that a_1 is the largest possible, that a_2 is the largest possible among the possible solutions that maximize a_1, and so on.

Input

The first line of input contains an integer N ($1 \leq N \leq 10^5$) and the second line contains N integers, separated by spaces, corresponding to the list $a_1, \ldots, a_N$ ($0 \leq a_i \leq 10^9$).

Output

Your program should print a single line containing N space-separated integers corresponding to the lexicographically maximum obtainable sequence.

Input example 1	**Output example 1**
4 8 4 2 1	15 0 0 0

Input example 2	**Output example 2**
4 12 15 1 20	31 13 4 0