Information

Memory limit

The limit is 512 MiB for each problem.

Source code limit

The size of each solution source code can't exceed 256 KiB.

Submissions limit

You can submit at most 50 solutions for each problem.

You can submit a solution to each task at most once per 30 seconds. This restriction does not apply in the last 15 minutes of the contest round.

Scoring

Each problem consists of several subtasks. The subtask score is awarded if all tests in the subtask are passed.

The number of points scored for the problem is the total number of points scored on each of its subtasks. The score for the subtask is the maximum number of points earned for this subtask among all the solutions submitted.

Feedback

To get feedback for your solution, go to "Runs" tab in PCMS2 Web Client and use "View Feedback" link. In each problem of the contest you will see the score for each subtask, or the verdict for the first failed test.

Scoreboard

The contestants' scoreboard is available during the contest. Use "Monitor" link in PCMS2 Web Client to access the scoreboard. The standings provided in PCMS2 Web Client are not final.

Problem A. The Battle of Giants

Time limit: 1 second

The famous programming contest organizer decided to hold competition for champions “The Battle of Giants”. There are two teams competing in the battle. Several matches are organized for the competition.

Each match can end with a win for one of the teams, or with a draw. When a team wins the match it scores 3 points, in this case the opposite team doesn’t score any point. In the case of a draw, teams score 1 point each. After all the matches end, the final score a:b is calculated: a and b — the number of points scored by the first and second team, respectively.

For example, if the first match is won by the first team, the second match ended with a draw, and the third match is also won by the first team, the final score is 7:1.

You are given the final score. Find out whether the given score is possible, and what is the minimum number of matches could be. Print the number of matches won by the first team, the number of matches ended with a draw, and the number of matches won by the second team.

Input

The first line contains a single integer a — the number of points scored by the first team.

The first line contains a single integer b — the number of points scored by the second team.

The given a and b are nonnegative and don’t exceed 10^9.

Output

If the given final score isn’t possible, print -1.

Otherwise, print three integers: the number of matches won by the first team, the number of matches ended with a draw, and the number of matches won by the second team, respectively. Find any solution with the minimum number of matches played.

Scoring

Subtask	Score	Constraints
1	21	$0 \le a \le 50$; $b = 0$
2	23	$0 \le a, b \le 50$; $a = b$
3	25	$0 \le a, b \le 50$
4	31	$0 \le a, b \le 10^9$

Examples

standard input	standard output
7 1	2 1 0
2 1	-1

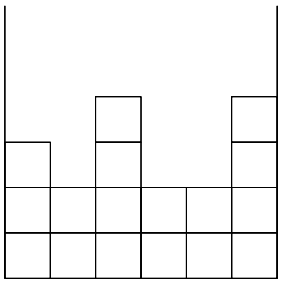
Problem B. Tetris Remastered

Time limit: 1 second

Mila likes to play Tetris. Today she learnt about a new game that is similar to Tetris. The new game has an infinite rectangular field with the bottom and width equal to n, divided into cells of size 1×1. Unlike real Tetris in this game horizontal pieces with height 1 and width x consisting of x cells — of size $1 \times x$, are used. Before the next piece starts to fall, a player can choose it's size x as any integer between 1 and n, inclusive. Pieces can't be rotated, but they can be moved left or right. A piece falls until it reaches an occupied cell under it or the bottom of the field.

Mile doesn't like to leave empty cells under the pieces. Her goal is to fill lower rows of the field in the way that all occupied cells form a rectangle of width n.

You are given a field state: $a_1, a_2, \ldots, a_n$, where a_i — the number of occupied cells in the i-th column of the field. In the given field no empty cell is under the occupied one. For example, if sequence a is $3, 2, 4, 2, 2, 4$, the field looks like this:



Find the minimum number of pieces Mila needs to play to occupy the lower rows of the field forming a rectangle of width n.

Input

The first line contains a single integer n — the width of the field ($1 \le n \le 2 \cdot 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ — the number of occupied cells in each column of the field ($0 \le a_i \le 10^9$).

At least on of the a_i is strictly greater than 0.

Output

Print a single integer: the minimum number of pieces Mila will need to build a rectangle of width n.

Scoring

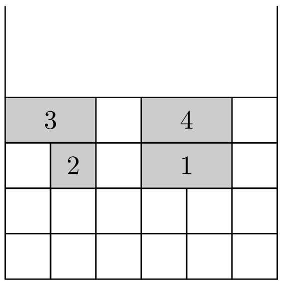
Subtask	Score	Constraints	
		n	a_i
1	8	$n \le 10$	$a_i \le 5$
2	13	$n \le 100$	$a_i \le 500$
3	16	$n \le 1000$	$a_i \le 5000$
4	17	$n \le 1000$	$a_i \le 10^9$
5	25	$n \le 10^5$	$a_i \le 10^9$ when $n > 1000$ sequence a is generated randomly
6	21	$n \le 2 \cdot 10^5$	$a_i \le 10^9$

Example

standard input	standard output
6 3 2 4 2 2 4	4

Explanation

In the example Mile can use the following four pieces:



Problem C. Optimal Truck

Time limit: 2 seconds

Peter is going to buy a truck and start a small transportation business. He studied the market and found out that there are n potential customers in his city. There are m_i contract options for the i-th client. Each option is specified by two numbers: w_{ij} — the minimum capacity of the truck, which is required to fulfill the contract, and c_{ij} — the profit that Peter will receive if he concludes this contract. No more than one contract can be concluded with each client.

Now Peter is thinking which truck is better to buy in order to get the profit he needs. He has q options. In the i-th option, Peter wants his profit to be at least x_i. Help him, for each of the options, find the minimum capacity of the truck with which you can make such a profit.

Input

The first line contains the integer n ($1 \le n \le 10^5$).

This is followed by n blocks describing contract options for each of the potential customers. Each such block begins with the number m_i, followed by m_i pairs of numbers w_{ij}, c_{ij} ($1 \le m_i$, $\sum m_i \le 5 \cdot 10^5$, $1 \le w_{ij}, c_{ij} \le 10^9$).

Next comes the number q ($1 \le q \le 10^5$). This is followed by the q numbers x_i ($1 \le x_i \le 10^9$).

Output

Print q numbers, the minimum carrying capacity of the truck, with which you can get the required profit, for each of the options. If it is impossible to get the required profit, print -1 for the corresponding option.

Scoring

Subtask	Score	Constraints			
		n	$\sum m_i$	q	Additional
1	7	$n \le 5$	$\sum m_i \le 10$	$q \le 10$	$w_i, c_i, x_i \le 100$
2	15	$n \le 100$	$\sum m_i \le 500$	$q \le 10$	$w_i, c_i, x_i \le 100$
3	21	$n \le 10^5$	$\sum m_i \le 5 \cdot 10^5$	$q \le 10$	—
4	24	$n \le 10^5$	$\sum m_i \le 5 \cdot 10^5$	$q \le 10^5$	$w_i \le 100$
5	33	$n \le 10^5$	$\sum m_i \le 5 \cdot 10^5$	$q \le 10^5$	—

Example

standard input	standard output

3
2
10 20
20 30
1
40 50
3
2 5
1 10
4 7
5
10 55 32 100 17
	1 40 20 -1 10

Problem D. Bookshelf Sorting

Time limit: 2 seconds

Irma works in a library. Every day she watches visitors take a couple of books from the bookshelf, read them, and put books back in the same places they took them. Usually people mess up the order and swap two books they read. Let's take a look at one specific bookshelf with n books in some order, numbered from 1 to n from left to right. The i-th visitor takes books from positions x_i and y_i and puts them back on the same positions, but in the wrong order. After the i-th visitor, the book that was at x_i is now at position y_i and vice versa.

In the evening, after the library is closed, Irma wants to put all the books back on their places. For each book there is a number p_i — a position where that book should end up in the end. To rearrange books, Irma can take any book from the shelf and insert it in the beginning or in the end (so it ends up in the first or in the last place on the shelf).

What is the minimum number of moves Irma can do to put all books in order? Answer this question for some initial placement of books, determined by p_i, and after each visitor that swapped places of some two books.

Input

The first line contains two integers n and q ($2 \le n \le 2 \cdot 10^5$; $0 \le q \le 2 \cdot 10^5$) — the number of books on the shelf and the number of visitors. The next line contains n distinct integers p_i ($1 \le p_i \le n$), meaning that the book in the i-th position must end up in position p_i.

Next q lines describe library's visitors. Each line contains two integers x_i, y_i ($1 \le x_i < y_i \le n$), that mean that the i-th visitor swapped two books on positions x_i and y_i.

Output

Print $q + 1$ integers — the minimum number of moves required to sort all books for the initial order, then for the order the after the first visitor, …, after all q visitors.

Scoring

Subtask	Score	Constraints
1	10	$n, q \le 8$
2	15	$n, q \le 200$
3	15	$n \le 2000$; $q = 0$
4	15	$n, q \le 2000$
5	20	$n \le 2 \cdot 10^5$; $q = 0$
6	25	$n, q \le 2 \cdot 10^5$

Example

standard input	standard output

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

Note

The initial order of books is $(5, 1, 2, 4, 3)$. To sort these books out, first, put the book 4 in the end, then do the same with book 5. After the first visitor, the bookshelf now looks like $(5, 1, 2, 3, 4)$; it's enough to just move the book 5 to the end. After the second visitor the order of books is $(3, 1, 2, 5, 4)$. For this order the minimum number of moves is 3, there are several ways to achieve the final order in 3 moves.

Problem E. Nice Shape

Time limit: 4 seconds

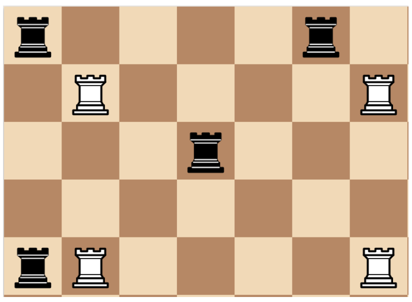
You are given n rooks on the different cells of the infinite chessboard.

The i-th of them is in the cell (r_i, c_i).

In one move you can move any rook to any cell in the same row/column. In other words, in one move you can choose any i and then either replace r_i to any other integer or replace c_i to any other integer. You can't move a rook to the cell with some other rook.

Four different rooks a, b, c, d form a *nice shape* if you can find a rectangle such that a, b, c, d are its corners. In other words, if the set of cells $\{(r_a, c_a), (r_b, c_b), (r_c, c_c), (r_d, c_d)\}$ is equal to the set of cells $\{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)\}$ for some integers x_1, x_2, y_1, y_2 with $x_1 \neq x_2$ and $y_1 \neq y_2$.

For example, the white rooks in the following picture form a nice shape.



Your goal is to find the minimum number of moves that you can perform to get a nice shape.

In other words, you need to find the minimum number of moves that you can perform, such that after them it will be possible to find a rectangle with four rooks in its corners.

Input

The first line of input contains one integer t ($1 \leq t \leq 25\,000$): the number of test cases.

The description of t test cases follows.

The first line contains one integer n ($4 \leq n \leq 100\,000$).

The i-th of the next n lines contains two integers r_i, c_i ($1 \leq r_i, c_i \leq 10^9$)

For each pair i, j with $i \neq j$, $r_i \neq r_j$ or $c_i \neq c_j$.

The total sum of n is at most $100\,000$.

Output

For each test case, print one integer: the minimum number of moves you need to perform to obtain at least one nice shape among given rooks.

Scoring

Subtask	Score	Constraints
1	10	$n \leq 4$
2	10	$n \leq 50$
3	10	$n \leq 200$
4	30	$n \leq 2000$
5	40	$n \leq 10^5$

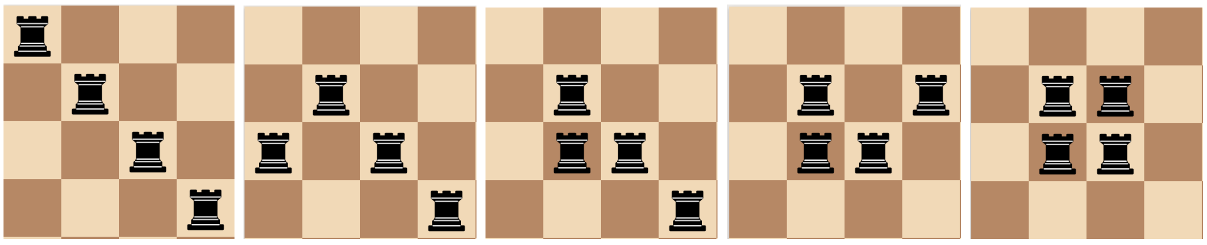
Example

standard input	standard output

5
4
4 4
1 1
2 2
3 3
4
4 4
4 1
1 4
2 2
6
3 2
2 1
1 2
3 3
3 4
3 1
5
1 1
1 2
1 3
1 4
5 5
4
1000000000 1000000000
1000000000 1
2 2
1000000000 999999999
	4
	2
	1
	3
	3

Note

One of the possible optimal solutions for the first test case of the example:



One of the possible optimal solutions for the second test case of the example: