

Problem A. Long Live

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

In Minecraft, there are three dimensions in the simple world, one of which is known as “The End”. In this dimension, the Ender Dragon is usually symbolized as the final boss of Minecraft.



Now we, you, and Steve want to beat the dragon. However, we find that the dragon is telling a puzzle. We must solve it before fighting against the dragon. Please stand by our side to fight against the dragon. The puzzle goes as follows.

Given two positive integers x and y , please find two integers a and b , such that

$$\sqrt{\frac{\text{lcm}(x, y)}{\text{gcd}(x, y)}} = a\sqrt{b}$$

and to maximize $a \cdot b$.

Here, $\text{gcd}(a, b)$ represents the greatest common divisor of a, b , while $\text{lcm}(a, b)$ represents the least common multiple of a, b .

Input

The first line contains an integer T ($1 \leq T \leq 10^4$) indicating the number of test cases.
Of the following T lines, each contains two integers x and y ($1 \leq x, y \leq 10^9$), representing the puzzle.

Output

Output T lines. Each contains two integers a and b , indicating the answer to each test case.

Example

standard input	standard output
2	1 1
1 1	1 1
4 4	

Note

Dear all participants:
This problem may be the very first page, but not where the story line ends. Please believe in yourself all the time, wishing you with my singing, “Long live all the mountains we moved, I had the time of my life fighting dragons with you”.

Problem B. Nana Likes Polygons

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

Nana’s electronic pet, MACARON, enjoys moving around in the room. Nana wants to create a bounded area where MACARON can roam within.

The positions in the room can be represented using a two-dimensional coordinate system, and Nana provides a set of vertices on it. Nana likes polygons and now wants to select a subset of vertices that can form a convex polygon as its endpoints. However, Nana also wants to ensure that the selected area is not too big.

Thus, she is going to determine the minimum possible area of the convex polygon ending on a subset of given vertices. If such a convex polygon doesn’t exist, please output “-1”(without quotes).

Input

The first line contains a single integer T ($1 \leq T \leq 10$) — the number of test cases.
For each test case, the first line contains a single integer n ($1 \leq n \leq 100$) — the number of given vertices.
The following n lines each contain two integers, and the i -th line contains x_i, y_i ($-1000 \leq x_i, y_i \leq 1000$) — the coordinate of the i -th vertex.

Output

For each test case, if no such convex polygon, please output “-1” without quotes; otherwise output a single real number denoting the minimum possible area of the convex polygon ending on these given vertices.
Let a and b denote the answer and your output respectively. Your output will be accepted if the absolute or relative error of b does not exceed 10^{-6} . Formally, your output is accepted if and only if $\frac{|a - b|}{\max(1, a)} \leq 10^{-6}$.

Example

standard input	standard output
2	2
4	-1
0 -1	
-3 0	
0 2	
2 2	
3	
-1 -1	
0 0	
1 1	

Note

For the first test case, choose $(2, 2)$, $(-3, 0)$, and $(0, 2)$ as endpoints of a convex polygon with area 2. It can be proved that this is the minimized area in this case.
For the second test case, no convex polygon can be formed by these three vertices.

Problem C. Lili Likes Polygons

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

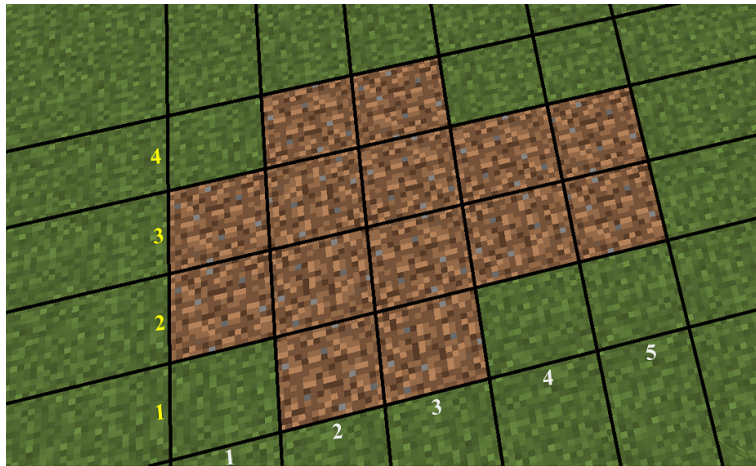
Lili and Nana are weeding in the backyard of Lili's house. They are repeatedly selecting rectangular areas and removing all the grass within them.

The backyard can be visualized as a 2D grid, each square cell representing one unit area. They have performed a total of n operations. During the i -th operation, they choose the left, bottom, right, and top sides of a rectangle, denoted as l_i, b_i, r_i, t_i , and clear all the cells within this rectangle using a lawnmower.

Note that these rectangles may overlap with each other.

Let $[l_i, r_i] \times [b_i, t_i]$ denote a rectangle.

Here is an example illustrated in the following figure. They have selected 2 rectangles, with the first rectangle being $[1, 5] \times [2, 3]$ and the second rectangle being $[2, 3] \times [1, 4]$.



After the n weeding operations, the union of the bare area may not be connected but all the sides are horizontal or vertical. Thus, the union becomes orthogonal polygon(s), some of which contain polygon holes. Moreover, there can be bare cells inner some holes. Please see the example inputs for more details and illustrations.

Now, they want to restore the land by planting some plants on the bare cells. Lili likes polygons, especially rectangles. Therefore, they want to select several rectangles, and these rectangles do not overlap with each other and exactly cover all the bare cells. Then, they plant different plants in different rectangles they selected.

For example, here is a feasible selection of rectangles for the aforementioned case: choose $[1, 1] \times [2, 3]$, $[2, 3] \times [1, 4]$ and $[4, 5] \times [2, 3]$.

After playing for a while, these two little girls have become tired, so they want to know the minimum number of non-overlapping rectangles that can cover all the bare cells.

Input

The first line contains a single integer n ($1 \leq n \leq 300$) — the number of rectangles when they weed.

The following n lines each contain 4 integers, and the i -th line contains l_i, b_i, r_i, t_i ($1 \leq l_i, b_i, r_i, t_i \leq 10^9, l_i \leq r_i, b_i \leq t_i$) — the left, bottom, right, and top sides of the i -th rectangle.

It is guaranteed the sum of the endpoints of the bare area (polygon(s) with polygonal holes) does not exceed 2000.

Output

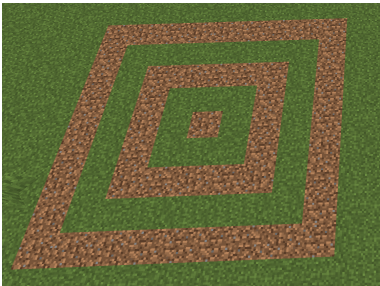
Output a single integer on a single line denoting the minimum number of rectangles they need to select to plant plants on all the bare cells.

Examples

standard input	standard output
8 1 1 1 1 1 2 1 2 1 3 1 3 2 1 2 1 2 3 2 3 3 1 3 1 3 2 3 2 3 3 3 3	4
2 1 1 100 100 1 501 100 600	2
4 1 1 4 1 1 4 5 4 1 1 1 4 4 1 4 5	5
9 1 1 9 1 1 1 1 9 1 9 9 9 9 1 9 9 3 3 7 3 3 3 3 7 3 7 7 7 7 3 7 7 5 5 5 5	9

Note

For the first example, an optimal selection is $[1, 1] \times [1, 3]$, $[2, 1] \times [2, 1]$, $[2, 3] \times [2, 3]$ and $[3, 1] \times [3, 3]$.
For the second example, an optimal selection is $[1, 1] \times [100, 100]$ and $[1, 501] \times [100, 600]$.
For the third example, an optimal selection is $[1, 1] \times [4, 1]$, $[1, 4] \times [5, 4]$, $[1, 2] \times [1, 3]$, $[4, 2] \times [4, 3]$ and $[4, 5] \times [4, 5]$.
For the fourth example, the bare area is illustrated in the following figure.



Problem D. MACARON Likes Happy Endings

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

MACARON is going to read a book, which contains n chapters, and the i -th chapter has a_i characters. MACARON wants to read this book in the next k days. On each day, he either reads several chapters from the first unread chapter, or just takes a rest (does not read the book), but he should finish his reading in k days.

MACARON enjoys his reading time and likes happy endings, so he does not want to be too sad during these days. He has an unlucky number d , because he thinks number d will lead to a bad ending. We use a sadness value to quantify his mood on each day. On the i -th day, if he reads, suppose that he reads chapters from L_i to R_i . The sadness value of this day is the number of intervals $[l, r]$ such that $L_i \leq l \leq r \leq R_i$ and $\bigoplus_{i=l}^r a_i = d$. The \oplus denotes the bitwise XOR operator. If he doesn't read, let the sadness value be 0.

MACARON wants to arrange his reading schedule to minimize the sum of sadness value in k days. Can you help him find the minimum value?

Input

The first line contains three integers n, k and d ($1 \leq n \leq 10^5, 1 \leq k \leq \min(n, 20), 0 \leq d \leq 10^6$) — the number of chapters, the days for reading, and the unlucky number.

The second line contains n integers a_i ($0 \leq a_i \leq 10^6$) — the number of characters in each chapter.

Output

Output a single integer denoting the minimized sum of the sadness value.

Example

standard input	standard output
10 4 5 1 2 3 4 5 5 6 5 4 3	3

Note

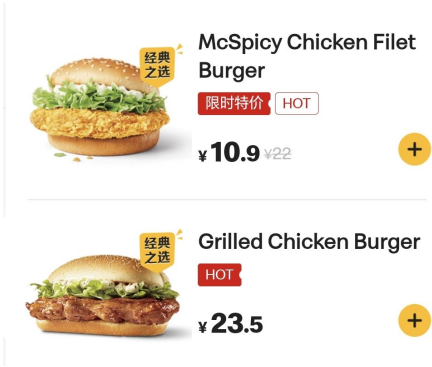
Here is an optimal schedule for reading:

- on the first day, just take a rest, and the sadness value is 0;
- on the second day, read from chapter 1 to chapter 3, and the sadness value is 0;
- on the third day, read from chapter 4 to chapter 7, and the sadness value is 2;
- on the fourth day, read from chapter 8 to chapter 10, and the sadness value is 1.

Problem E. Spicy or Grilled?

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Since a regular programming contest lasts for five hours, it's a big problem to prepare contestants' food during the contest.



In the 20XX International Collegiate Programming Contest in Hubei Province, China, there were n contestants participating in the contest. The host wanted to prepare a Spicy Chicken Burger Set for every contestant, which was a dollars per set at that time. But since there were some people who could not bear the spiciness like Walk Alone, a Grilled Chicken Burger Set was prepared for these people, which was b dollars per set. Before the contest, the host had collected the number x of people who wanted to eat a Grilled Chicken Burger Set.

But Walk Alone was too stupid and lazy to calculate how much the host would spend. Please help him to calculate.

Input

In the first line, there is an integer T ($1 \leq T \leq 10^4$) denoting the number of test cases. For each test case, there are four integers n , x , a and b ($1 \leq n, a, b \leq 10^4, 0 \leq x \leq n$) in each line, representing the number of contestants in total, the number of contestants who want Grilled Chicken Burger Set, the price of Spicy Chicken Burger Set and Grilled Chicken Burger Set per set.

Output

For each test case, output a single integer denoting the total money the host would spend.

Example

standard input	standard output
3	16000
600 200 27 26	19500
750 0 26 27	750
750 750 1 1	

Note

In the first test case, there were 400 people chosen the Spicy Chicken Burger Set and 200 people chosen the Grilled one. So the total money is $400 \times 27 + 200 \times 26 = 16\,000$ dollars.

Problem F. Enchanted

Input file: **standard input**
Output file: **standard output**
Time limit: **1.5 seconds**
Memory limit: **1024 megabytes**

In Minecraft, one way to become stronger is to have the armors and weapons enchanted. Enchanted books play an important role.



The most important attribute of an enchanted book is its level. The higher the level is, the better the book is. We could merge two books with the same level l into one new book (the older two will disappear). The level of the new book is $l + 1$ and the merger costs 2^{l+1} .

Now, Steve has n enchanted books numbered from 1 to n . Initially, the i -th book's level is a_i . Steve asks you to help him with the following four operations.

1. Given two integers l, r ($1 \leq l \leq r \leq n$), calculate the maximum level Steve can reach by merging books numbered from l to r .
2. Given three integers l, r ($1 \leq l \leq r \leq n$) and k , then follow the steps:
Step 1: Steve merges all the books numbered from l to r until there does not exist two books with the same level.
Step 2: Steve adds one new book with level k to the books he gets in Step 1.
Step 3: Steve needs to merge books he gets in Step 2 and he wants to maximize the times he merges the books.
Please calculate and output the total cost modulo $10^9 + 7$ in Step 3.
Note that, after calculating, the sequence is restored. That is, this operation does NOT actually change the sequence.
3. Given two integers pos, k , Steve changes the level of the book numbered pos to k .
4. Given one integer t , Steve restores the sequence to the state after the t -th operation. If $t = 0$, then Steve restores the sequence to the beginning state.

Input

The first and the only line contains 5 integers n, m, A, p, q . ($1 \leq n \leq 10^6, 1 \leq m \leq 10^6, 1 \leq A \leq 19\,260\,817, 1 \leq p \leq 4, 1 \leq q \leq 30$)

Please pay attention! To avoid large input file, the true input is constructed through the following strategy:

Function `rnd()`:

```
A ← (7A + 13) mod 19 260 817  
return A
```

Firstly, n integers a_1, a_2, \dots, a_n are generated in turn, $a_i \leftarrow \text{rnd}() \bmod q + 1$.

Then, the parameters of all operations are generated.

For each operation, the number of operation (representing by opt) is firstly generated, $opt \leftarrow \text{rnd}() \bmod p + 1$.

According to the number of operation, the different method is applied to generate parameters for different operation.

- For operation 1: we need to get l and r by using:

$$\begin{aligned} L &\leftarrow rnd() \bmod n + 1 \\ R &\leftarrow rnd() \bmod n + 1 \\ l &\leftarrow \min(L, R) \\ r &\leftarrow \max(L, R) \end{aligned}$$

- For operation 2: we need to get l, r and k by using:

$$\begin{aligned} L &\leftarrow rnd() \bmod n + 1 \\ R &\leftarrow rnd() \bmod n + 1 \\ l &\leftarrow \min(L, R) \\ r &\leftarrow \max(L, R) \\ k &\leftarrow rnd() \bmod q + 1 \end{aligned}$$

- For operation 3: we need to get pos and k by using:

$$\begin{aligned} pos &\leftarrow rnd() \bmod n + 1 \\ k &\leftarrow rnd() \bmod q + 1 \end{aligned}$$

- For operation 4: we need to get t by using:

$$t \leftarrow rnd() \bmod i$$

Here, i represents the i -th operation.

Output

For each operation 1 and 2, you need to output one single line with an integer, representing the answer to operation 1 and 2 modulo $10^9 + 7$.

Examples

standard input	standard output
6 3 2 1 3	1 3 2
10 15 5 4 7	0 9 9 0 64 0 0

Note

Function “max” means the maximum one between the parameters. Function “min” means the minimum one between the parameters.

In example 1, the initial books are $[1, 2, 3, 1, 2, 3]$. The three operations’ ranges are $[4, 4]$, $[1, 3]$ and $[4, 5]$.

Problem G. Genshin Impact Startup Forbidden II

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 512 megabytes



Blue-edged Shot is forbidden from playing Genshin Impact by LeavingZ, so she turned to Go game.

A game of Go involves two players, one playing with Black stones and the other with White stones. Two players take turns making moves, with the Black stones going first. The Go board is composed of a grid of 19×19 intersections, and we use (x, y) to represent the intersection at the x -th row and y -th column. The stones are placed on the intersections. The top left corner is $(1, 1)$, while the bottom right corner is $(19, 19)$.

The intersections (x_1, y_1) and (x_2, y_2) are adjacent if and only if $|x_1 - x_2| + |y_1 - y_2| = 1$. Adjacent intersections with stones of the same color belong to the same group of stones. The number of “liberties” for a stone is equal to the number of adjacent intersections to the stone’s intersection that do not have any stones on them. The liberties of a group of stones equal the sum of the liberties of all the stones belonging to that group. A group of stones with zero liberties is considered dead and must be removed from the board.

Note that after Black plays, priority is given to removing any dead White stones, and then recalculating the liberties for Black stones. This is because there might be situations where, after Black plays, both Black and White stones have zero liberties, but removing the dead White stones increases the liberties for Black stones. As for White, process the stones similarly. After White plays, priority is given to removing any dead Black stones, and then recalculating the liberties for White stones.

Now there is a game of Go, starting with an empty board, and a total of m moves have been made by both players. Given the positions where each stone is placed, output after each move, how many Black and White stones are removed respectively causing by this move. Obviously, black stones are played on odd-numbered moves, and white stones are played on even-numbered moves. It’s guaranteed that stones are placed on empty intersections. Note that stones can be placed on **any** intersections that do not have stones on them at the moment, regardless of violating the Go rules in real life.

Input

Input m ($1 \leq m \leq 5 \times 10^5$) lines, the i -th line contains two integers x_i, y_i ($1 \leq x_i, y_i \leq 19$), representing the i -th move puts stone on (x_i, y_i) .

It’s guaranteed that stones are placed on intersections that do not have stones on them at the moment.

Output

Output m lines, each line contains two integers. The first integer in the i -th line represents the number of Black stones removed after the i -th move, while the second for White stones.

Example

standard input	standard output
8	0 0
2 1	0 0
1 1	0 1
1 2	0 0
2 2	0 0
1 1	0 0
1 3	0 0
2 3	3 0
3 1	

Note



Problem H. Genshin Impact Startup Forbidden III

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

Blue-edged Shot is forbidden from playing Genshin Impact by LeavingZ. However, today LeavingZ went to Huazhong University of Science and Technology, School of Cyber Science and Engineering, to harvest the gold medal of the 2024 International Collegiate Programming Contest in Hubei Province, China.

An activity of Genshin Impact called “Dodoco’s Bomb-Tastic Adventure” has begun. This is a single-player game, where each game involves a pond. The pond can be divided into an n by m grid, with the cell in the i -th row and j -th column represented by (i, j) . Among these, k cells contain fish, and you will play as the Spark Knight Klee, who fishes with explosives.



If Klee drops a bomb at (a, b) , all the cells (x, y) satisfying $|x - a| + |y - b| \leq 1$ will be covered by the explosion. For every cell covered by the explosion, Klee will catch one fish from it. Klee can drop bombs at any location. The question is, to catch all the fish, how many “Jumpy Dumpty” bombs are needed at a minimum?

Input

The first line contains three integers n, m, k ($1 \leq n, m \leq 10^3, 1 \leq k \leq 10$) — the size of the grid and the number of cells containing fish.

The following k lines, each line contains three integers x_i, y_i, a_i ($1 \leq x_i \leq n, 1 \leq y_i \leq m, 1 \leq a_i \leq 3$), representing there are a_i fish in cell (x_i, y_i) .

It is guaranteed that the input cell (x_i, y_i) are unique.

Output

Output a single integer, denoting the minimum number of bombs needed.

Example

standard input	standard output
5 5 3 1 1 2 2 2 1 5 5 2	4

Note

One possible way is to drop two bombs at $(1, 2)$ and another two at $(5, 5)$.

It can be proven that there is no solution with a smaller answer.

Problem I. Colorful Tree

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 512 megabytes

You are given a tree with n nodes numbered from 1 to n and $n - 1$ edges. Each node has a color. Initially, all of them are white.

We are going to perform q operations. In each operation, two vertices u and v will be given, and we will color black to the points along the simple path from u to v (u, v inclusive). Note that a simple path in the tree is defined as a path that does not pass through any vertex more than once.

After each operation, you are required to determine the length of the longest simple path in the tree where all nodes on the path are the same color. The length of a path is defined as the number of nodes on the path.

Input

The first line contains a single integer T ($1 \leq T \leq 100$) indicating the number of test cases.

For each test case, the first line contains two integers n ($1 \leq n \leq 2 \times 10^5$) and q ($1 \leq q \leq 2 \times 10^5$), indicating the number of nodes in the tree and the number of operations, respectively.

In the following $n - 1$ lines, each contains two integers u and v , representing an edge from vertex u to v in the tree.

Then follow q lines, each contains two integers u and v , representing an operation that colors black to the points along the path from vertex u to v .

It is guaranteed that the sum of n and q of all the test cases in a test does not exceed 2×10^5 respectively.

Output

For each test case, output q lines, each line should contain an integer representing the length of the longest simple path in the tree where all nodes on the path are the same color after the corresponding operation.

Example

standard input	standard output
1	5
8 6	4
1 2	4
1 3	3
2 4	4
4 5	4
2 6	
4 8	
3 7	
4 8	
7 7	
4 5	
2 2	
4 6	
5 1	

Problem J. Points on the Number Axis A

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Alice is playing a single-player game on the number axis.

There are n points on the number axis. Each time, the player selects two points. The two points will be removed, and their midpoint will be added. When there is only one point on the axis, the game ends. Formally, if the two chosen points is x_i and x_j , then $\frac{x_i + x_j}{2}$ will be added after the operation.

In order to play this game happily, Alice will always select two points randomly.

Now Alice have a question: where is the expected position of the last point.

It can be proven that the answer can be expressed in the form $\frac{p}{q}$, you only need to output the value of $p \cdot q^{-1} \bmod 998\,244\,353$.

Input

The first line contains one integer n ($1 \leq n \leq 10^6$).

The second line contains n integers x_i ($0 \leq x_1 \leq \dots \leq x_n < 998\,244\,353$), denoting the position of the i -th point.

Note that two points may be in the same position.

Output

Output one integer, the answer modulo 998 244 353.

Example

standard input	standard output
3 1 2 4	332748120

Problem K. Points on the Number Axis B

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Bob is playing a single-player game on the number axis.

There are n points on the number axis. Each time, the player selects two points. The two points will be removed, and their midpoint will be added. When there is only one point on the axis, the game ends. Formally, if the two chosen points is x_i and x_j , then $\frac{x_i + x_j}{2}$ will be added after the operation.

In order to play this game happily, Bob will always select two adjacent points randomly. Initially, the i -th point and the $(i + 1)$ -th point are adjacent. When Bob added a new point, it inherits its left point's left adjacent point and its right point's right adjacent point.

Now Bob have a question: where is the expected position of the last point.

It can be proven that the answer can be expressed in the form $\frac{p}{q}$, you only need to output the value of $p \cdot q^{-1} \bmod 998\,244\,353$.

Input

The first line contains one integer n ($1 \leq n \leq 10^6$).

The second line contains n integers x_i ($0 \leq x_1 \leq \cdots \leq x_n < 998\,244\,353$), describing the position of the i -th point.

Note that two points may be in the same position.

Output

Output one integer, the answer modulo 998 244 353.

Example

standard input	standard output
3 1 2 4	623902723

Problem L. LCMs

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Walk Alone has a number axis with only positive integers on it. The cost of walking from point on integer a to point on integer b is $\text{lcm}(a, b)$, where $\text{lcm}(a, b)$ means the least common multiple of integers a and b . Due to the hatred of the integer 1, Walk Alone forbids anyone from moving to points on integers **smaller than or equal to 1**.

Given two integers a and b , you need to calculate the minimal cost of walking from point on integer a to b .

Input

There are T ($1 \leq T \leq 1000$) test cases.

In each test case, there is only one line containing two integers a and b ($2 \leq a \leq b \leq 10^7$), denoting the start and end point.

Output

For each test case, output a single integer denoting the minimal cost.

Example

standard input	standard output
3	10
3 4	25
10 15	4
2 4	

Note

In the first test case, you can walk such a path: $3 \rightarrow 2 \rightarrow 4$, where the total cost is $\text{lcm}(3, 2) + \text{lcm}(2, 4) = 6 + 4 = 10$, which can be proved to be the minimum.