Problem A. Array Counting

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Orange`

You are given three integers: N, S, and G.

We want you to count the number of arrays of length N that you can form; such that each array satisfies the following conditions:

- All elements of the array are positive integers.
- The sum of all elements is S.
- The GCD between all elements is G.
- The difference between the maximum and minimum elements is as minimum as possible.

Print the answer $mod\ 10^9 + 7$.

Input

The first line of input contains a single integer T — the number of test cases.

For each test case:

- The input contains three space-separated integers: N, S, and G ($1 \le N \le 10^6$, $1 \le S \le 10^{18}$ and $1 \le G \le 10^9$).

The sum of N for all test cases will not exceed 10^6.

Output

Print one integer — the answer to the problem $mod\ 10^9 + 7$.

Example

standard input	standard output

3
3 3 1
2 6 2
10 19 3
	1
	2
	0

Note

In the first test case, there's only one array with size = 3, sum = 3 and GCD = 1: $\{1,\ 1,\ 1\}$. The difference between the max and min elements of the array is 0.

In the second test case, there are two arrays with size = 2, sum = 6 and GCD = 2: $\{2,\ 4\}$, $\{4,\ 2\}$. The difference between the max and min elements of the both arrays is 2.

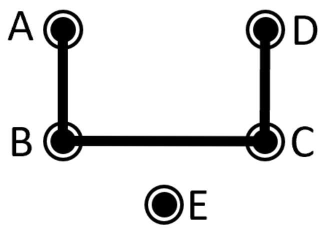
In the third test case, there are zero arrays with size = 10, sum = 19 and GCD = 3.

Problem B. Baa Damma Boo

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Red`

Some Arabic letters resemble geometric figures. For example, the letter ب -pronounced baa- looks like an open rectangle with a dot underneath.

Given a set of points on a 2D grid, count how many ب characters can you form. The character is composed of 5 points A, B, C, D and E. Four points form a rectangle $ABCD$. The rectangle doesn't have to align with the axes. The fifth point E must lie strictly between the two straight lines $\overline{AB}$ and $\overline{CD}$, and must lie under line segment $\overline{BC}$ from the opposite direction of $\overline{AD}$.



The Arabic letter ب

Input

The first line contains an integer T — the number of test cases.

The first line of every test case has a single integer N $(1 \le N \le 400)$ — the number of points.

N lines follow, each has a pair of space-separated integers $(X_i,\ Y_i)$ $(-10^6 \le X, Y \le 10^6)$ — the X and Y coordinates of the corresponding point, respectively.

All points are **distinct**.

The sum of N for all test cases will not exceed 400.

Output

For each test case, print one line containing a single integer — the number of occurrences of the letter ب on the grid.

Example

standard input

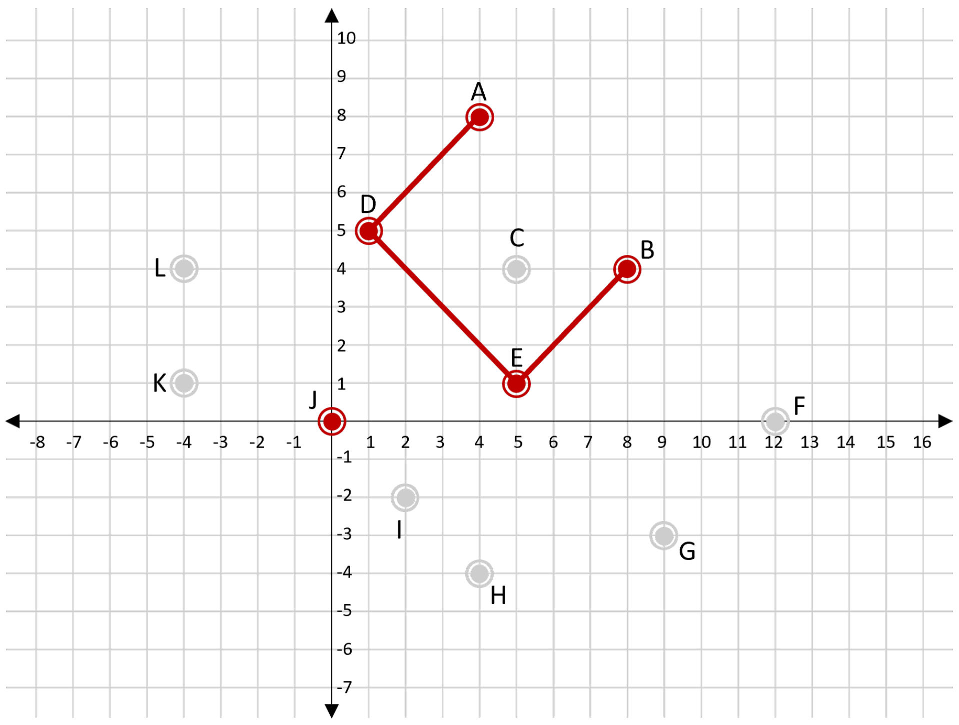
```
1
12
1 5
4 8
5 1
8 4
5 4
9 -3
12 0
0 0
2 -2
4 -4
-4 1
-4 4
```

standard output

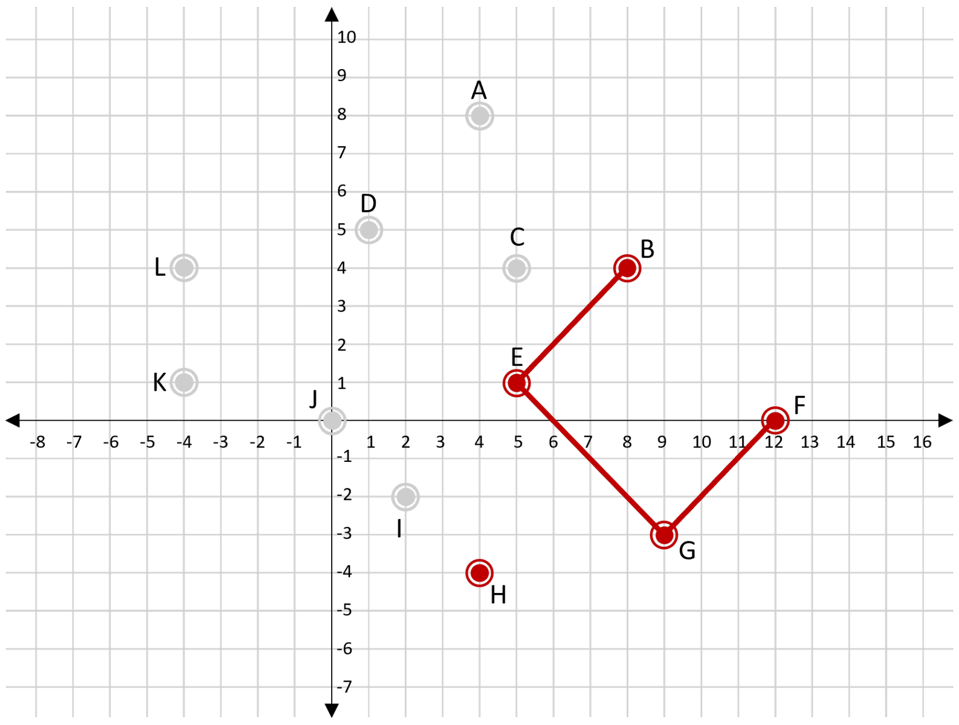
```
11
```

Note

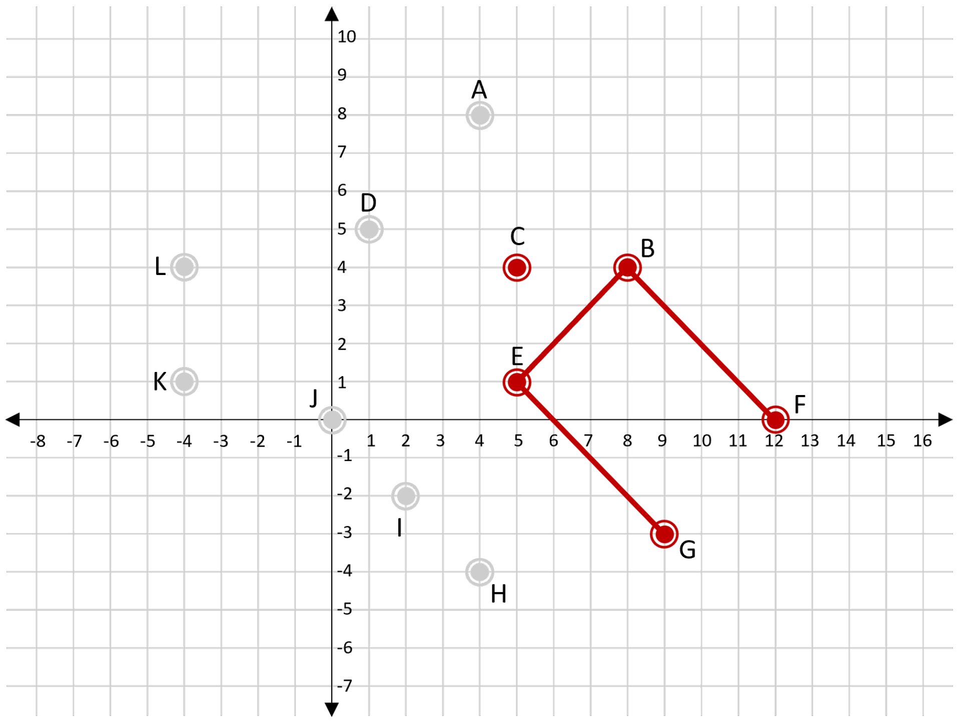
An illustration of the provided test case:



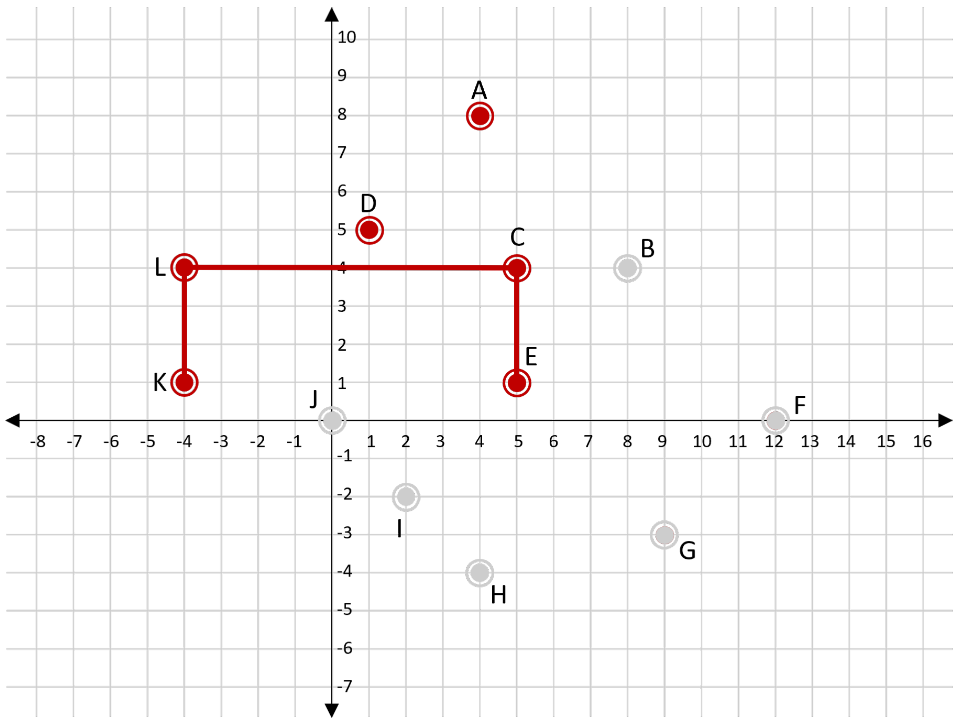
|___| $ADEB$ forms one occurrence of ب, with the point J.



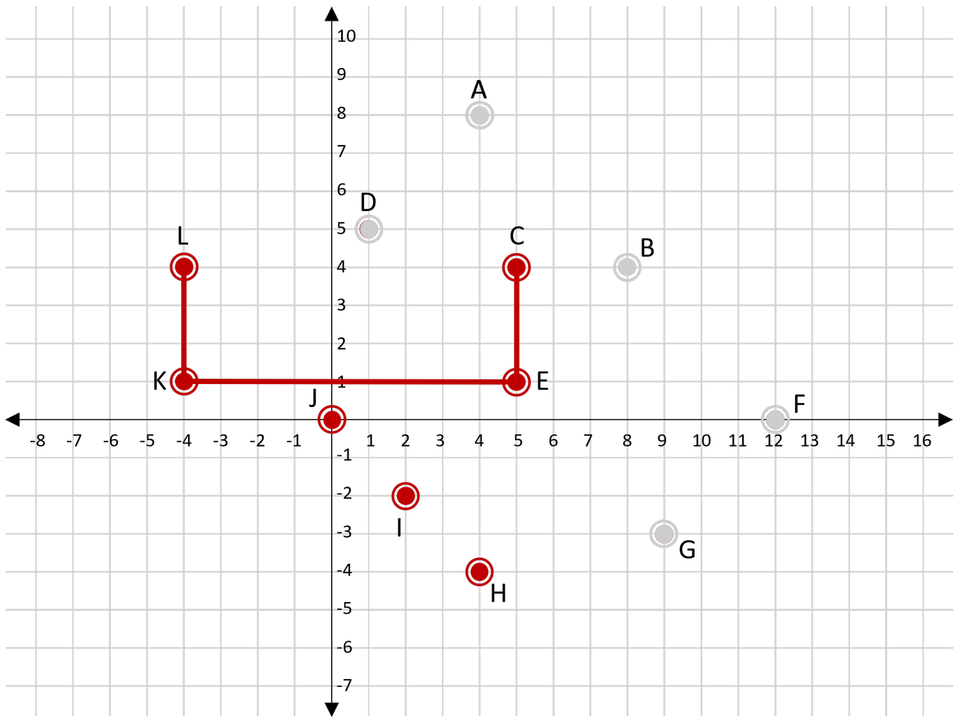
|___| $BEGF$ forms one occurrence of ب, with the point H.



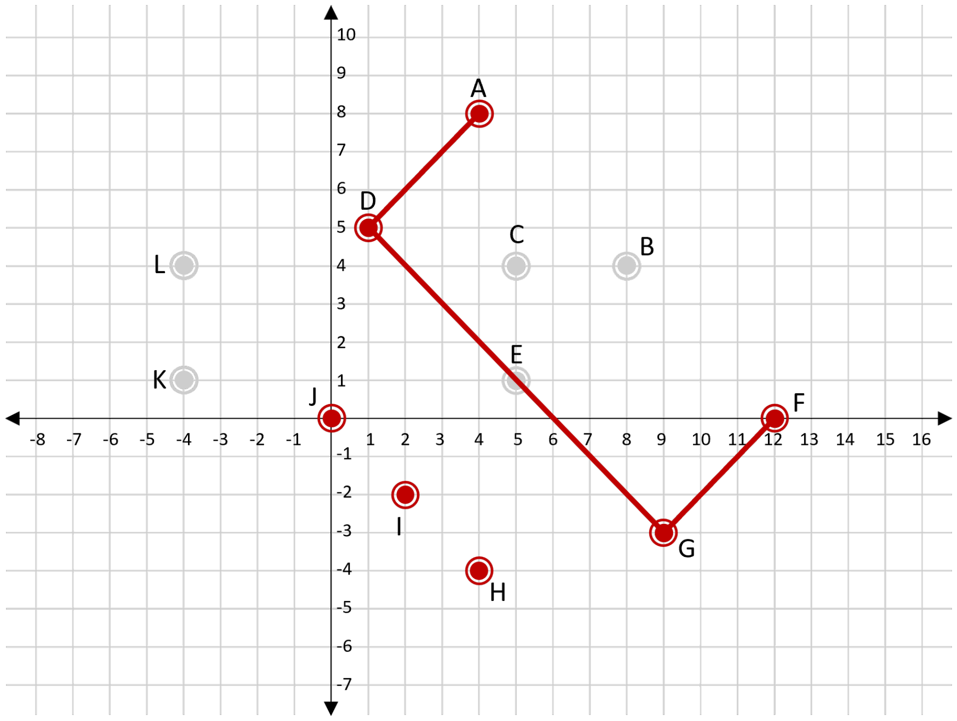
|___| $FBEG$ forms one occurrence of ب, with the point C.



|___| $ECLK$ forms two occurrences of ب, with each point from $[D,\ A]$.



|___| $LKEC$ forms three occurrences of ب, with each point from $[J,\ I,\ H]$.



|___| $ADGF$ forms three occurrences of ب, with each point from $[J,\ I,\ H]$.

Problem C. Cutting Edges

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Dark Blue`

You are given an undirected weighted connected graph of N nodes and M edges, where each edge between nodes A and B has a weight W_{AB}, and each node i has a value V_i.

Process Q queries; each is one of the following two types:

- $1\ X\ V^{\text{new}}$ – Change the value of node X to the new value V^{new}. (i.e) Set $V_X = V^{\text{new}}$.
- $2\ X\ Z$ – Print the maximum weight $W^{\max}$, such that if we cut all edges **strictly greater** than $W^{\max}$, the sum of V_i for all the nodes that are reachable from node X is $\leq Z$.
  - If Z is bigger than the sum of values of all nodes on the graph, print -1.
  - It's guaranteed that $Z \geq V_X$.

Input

The first line contains two space-separated integers N and M $(1 \leq N \leq 10^5,\ 1 \leq M \leq 10^5)$ — the number of nodes and edges, respectively.

The second line contains N space-separated integers V_i $(0 \leq V_i \leq 10^9)$ — V_i is the value of node i.

Then, M lines follow, each contains three space-separated integers A, B, and W_{AB} $(1 \leq A, B \leq N,\ 1 \leq W_{AB} \leq 10^9)$. **All W_i values are distinct**.

The next line contains a single integer Q $(1 \leq Q \leq 10^5)$ — the number of queries.

Q lines follow, each is one of the two types:

- $1\ X\ V^{\text{new}}$ $(1 \leq X \leq N,\ 0 \leq V^{\text{new}} \leq 10^9)$ — a query of type 1.
- $2\ X\ Z$ $(1 \leq X \leq N,\ Z \leq 10^{18})$ — a query of type 2.

Output

For each query of type 2, print $W^{\max}$ on a new line.

Example

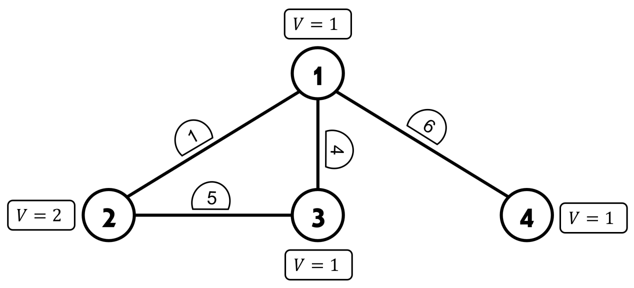
standard input	standard output

4 4
1 2 1 1
1 2 1
2 3 5
4 1 6
3 1 4
3
2 1 3
1 2 1
2 2 3
	3
	5

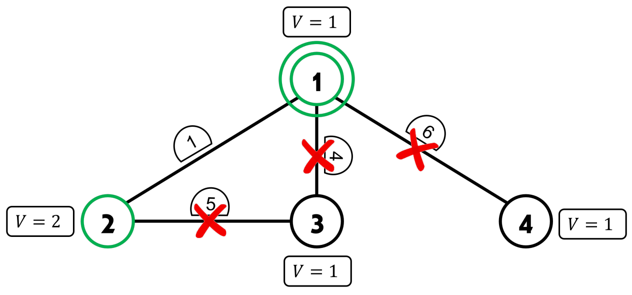
Note

Illustration of the example:

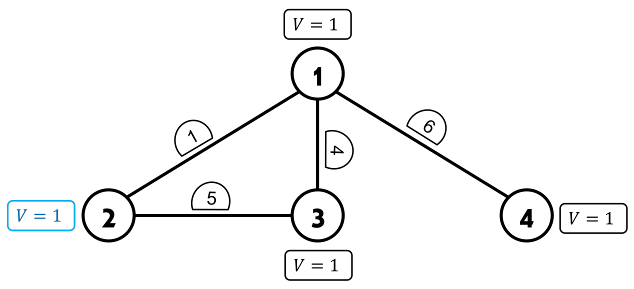
The graph **before** any queries:



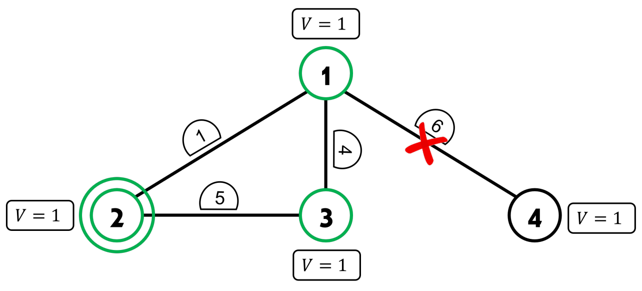
Query #1:



Query #2:



Query #3:



▢ Value of a node. ◠ Weight of an edge. ◎ Node X. ○ Nodes reachable by node X. ✗ Edges to be cut. ▢ New value of a node.

Problem D. ! Divisible

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Yellow`

You have an undirected tree of N nodes, each node i has a unique integer value A_i assigned to it.

You will then be given Q queries, where each query i has two integers U_i and V_i. You are asked to find the minimum value **(greater than 1)** that is not divisible by any value of any of the nodes on the path between U_i and V_i, including the nodes U_i and V_i themselves.

Input

The first line of input contains T — the number of test cases.

For each test case:

- The first line contains a single integer N $(1 \le N \le 2 \cdot 10^5)$ — the number of nodes.
- The next line has N unique values A_i $(2 \le A_i \le 10^6)$ — the value of each node.
- Each of the next $N - 1$ lines has a space-separated pair of integers: $(U_i,\ V_i)$ $(1 \le U_i, V_i \le N$ and $U_i \ne V_i)$ — which means there is an edge between U_i and V_i.
- The next line has a single integer Q $(1 \le Q \le 2 \cdot 10^5)$ — the number of queries.
- Each of the following Q lines has a space-separated pair of integers: $(U_i,\ V_i)$ $(1 \le U_i, V_i \le N$ and $U_i \ne V_i)$.

The sum of N for all test cases will not exceed $2 \cdot 10^5$.

The sum of Q for all test cases will not exceed $2 \cdot 10^5$.

It is guaranteed that the nodes will be connected to form a tree.

Output

Print Q lines, each has the answer for each query.

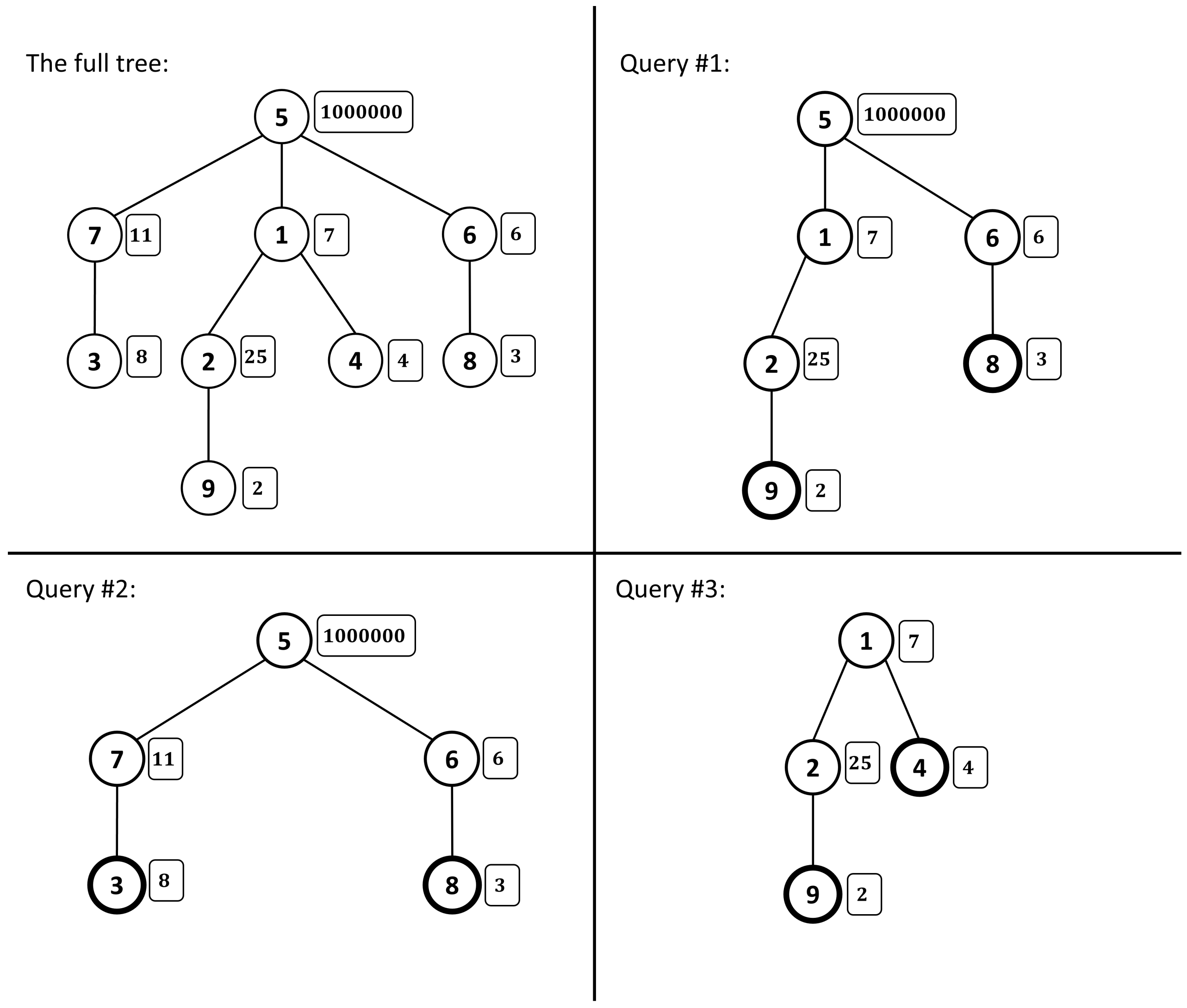
Example

standard input	standard output

1
9
7 25 8 4 1000000 6 11 3 2
5 7
5 1
5 6
7 3
1 2
1 4
6 8
2 9
3
8 9
3 8
4 9
	5
	2
	3

Note

An illustration of the test case:



Problem E. Erasing The Array

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Green`

You are given arrays A and B of N elements. You are allowed to do the following operations on the arrays:

- Remove the leftmost elements of both arrays, with index l, for the cost A_l. This can be done any number of times if the arrays are non-empty (have at least one element).
- Remove the rightmost elements of the arrays, with index r, for the cost B_r. This can be done any number of times if the arrays are non-empty (have at least one element).
- Remove both the leftmost and rightmost elements, with indices l and r respectively, for a fixed cost X, which is given. This operation can only be done K times, and only if each array has at least two elements left.

  If each array has only one element left, then the cost of removing it from both arrays is $\min(A_i,\ B_i)$.

Your task is to print the minimum cost to clear both arrays (remove all elements using one of the three operations above at each step).

Input

The first line of input contains three space-separated integers N, K, and X ($1 \leq N \leq 3 \cdot 10^5$, $0 \leq K \leq \lfloor N/2 \rfloor$ and $1 \leq X \leq 10^9$).

The second line contains N space-separated integers A_i ($1 \leq A_i \leq 10^9$) — the elements of array A.

The third line contains N space-separated integers B_i ($1 \leq B_i \leq 10^9$) — the elements of array B.

Output

Print the minimum cost required to clear the arrays.

Example

standard input	standard output

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

Problem F. Flowers Need Water

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Purple`

We have N barriers, modelled by lines on a 2D plane, none of which is vertical or horizontal.

Let's define a node as the intersection of two or more barriers. A flower is planted on each node.

We want our flowers to grow, and for that, they need water. We can only pour water on the topmost node, relative to the $Y-$ axis (**it's unique**).

We have a branching tool, that allows the water at a node to spread to its neighboring regions. Only the top most node has this tool right now.

When we pour water to the top most node, the flower in this node absorbs it, and water enters all regions around it. A region is an area surrounded by barriers. Water also flows to all other nodes touching those regions, so a flower in one of those nodes will immediately absorb the water and grow. However, if a node does not have a branching tool, the water does not go further, and does not flow to any of its neighboring regions.

For the water to flow from a node to all its neighboring regions, we have to install a branching tool on this node.

We want to know, for each node **independently**, what is the minimum number of branching tools that we have to install on the other nodes so that water reaches it. Print the sum of those numbers (the total for all nodes).

Input

The first line of input contains one integer N $(2 \le N \le 50)$ — the number of barriers.

N lines follow, each contains two space-separated integers A_i and B_i $(-1000 \le A_i, B_i \le 1000$ and $A_i \ne 0)$ — $(y = A_i \cdot x + B_i)$ is the equation of the i^{th} barrier.

All barriers are unique. Not all coefficients A will be equal; it is guaranteed there will be at least one node.

Output

One line that contains one integer — the sum of the minimum number of branching tools needed per node.

Examples

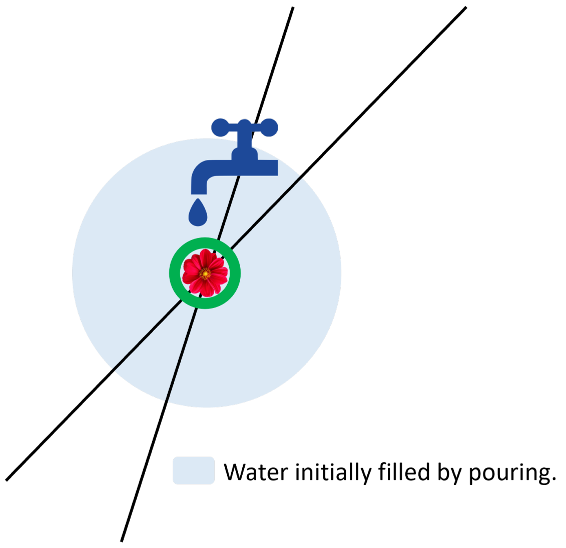
standard input	standard output

2
1 2
3 4
	0

4
6 70
-1 70
1 30
1 0
	2

Note

Below is an illustration of example 1:



Below is an illustration of example 2:

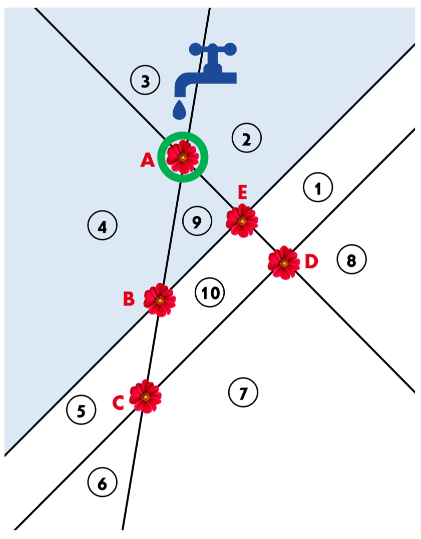


Figure 1

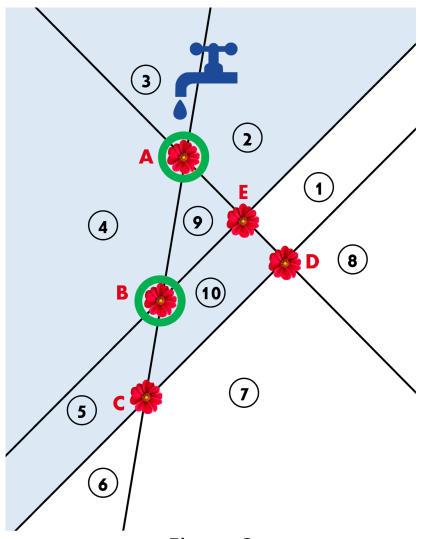


Figure 2

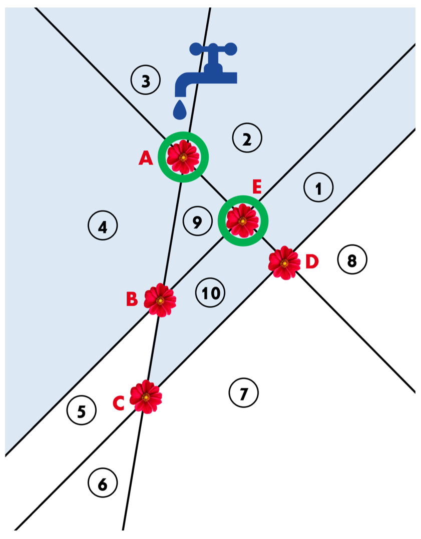


Figure 3

Figure 1 is an illustration of how water spreads into regions 2, 3, 4, 9, and nodes A, B and E after being poured on the topmost node A, which has a branching tool.

As you can see, nodes A, B and E do not need any branching tools to receive water (the branching tool at node A is by default), so the answer for these nodes is **0**.

For node C to get water, we can install **1** branching tool on node B -as in figure 2- so that water spreads into regions 5 and 10 and touches nodes C and D. Alternatively, we can install **1** branching tool on node E -as in figure 3- so that water spreads into regions 10 and 1 and touches nodes C and D. Since both alternatives require **1** branching tool, the answer for node C is **1**.

For node D, the options are exactly like the options mentioned for node C (installing a branching tool on node B or E). Each requires **1** branching tool, so the answer for node D is **1**.

The total answer becomes $0 + 0 + 1 + 1 + 0 = 2$.

Problem G. Game on a Board

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Bronze`

You are playing a game with your friend. You have Z zeros and O ones written on a board. In one turn, a player can select two numbers and replace them with 0 if they are similar, and 1 if they are different.

You win if the final number is 0, and your friend wins if the final number is 1. You always start playing, and you both play optimally.

Find out who will win.

Input

The first line of input contains T $(1 \le T \le 10^5)$ — the number of test cases.

For each test case:

- The input contains two space-separated integers Z and O $(0 \le Z, O \le 10^{18},\ 0 < Z + O)$ — the number of zeros and ones, respectively.

Output

For each test case, print one line containing a string; "First" (if you win) or "Second" (if your friend wins), without the quotes.

Examples

standard input	standard output
1 2 2	First
1 2 3	Second

Problem H. Hash Values

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Black`

You are given an array $A = \{A_0,\ A_1,\ ...,\ A_{N-1}\}$ of size N, where $N = 2^L$.

You are also given three integers $X,\ Y$ and M, and a list of K pairs of integers $(C_i,\ D_i)$.

Now, let's call a permutation P of size N good if: for all the given K pairs, these two conditions hold:

- The value C_i comes before the value D_i in that permutation.

  For example, if the the list of pairs is $\{(1,\ 3),\ (0,\ 3)\}$ and $N = 4$, the permutation $P = \{1,\ 0,\ 3,\ 2\}$ is good but the permutation $P = \{1,\ 2,\ 3,\ 0\}$ is not (because the value 0 is after the value 3).

- For every pair of indices ($i,\ j$ and $i \neq j$), we have the following condition satisfied: $\frac{1}{2} < \frac{P_i\ \oplus\ P_j}{i\ \oplus\ j} < 2$, where $\oplus$ is the bitwise exclusive-or operator.

Let's define the following hash function on the permutation $P = \{0,\ 1,\ ...,\ N-1\}$ of size N, as:

$$H(P) = (\textstyle\sum_{i=0}^{N-1} A[P[i]] \times X^{\text{pop\_count}(i)} \times Y^{L-\text{pop\_count}(i)})\ mod\ M$$

where pop_count(i) is the number of one bits in the binary expansion of i.

Find all the possible hash values **after the mod** for all the good permutations that can be constructed, given the list of K pairs, the array A and the coefficients X, Y and M.

Input

The first line of input contains five space-separated integers $L,\ X,\ Y,\ M$ and K ($1 \leq L \leq 10$, $0 \leq X,\ Y,\ K \leq 1000$ and $1 \leq M \leq 1000$).

The second line contains N space-separated integers A_i ($0 \leq A_i \leq 10^9$) — the elements of the array.

K lines follow, each line contains a space-separated pair of integers: $(C_i,\ D_i)$ ($0 \leq C_i < D_i \leq N-1$).

Output

On the first line, print one integer — the number of possible hash values.

On the second line, print the hash values (after the mod) separated by spaces in **any order**.

Example

standard input	standard output

`2 11 29 991 2`
`34 120 107 29`
`1 3`
`0 3`
	`4`
	`179 461 758 873`

Problem I. ICPCs In Egypt

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Pink`

Due to the pandemic, the ICPC schedule has been shifted multiple times. The 2020 Moscow ICPC was actually held in 2021, and the 2021 Bangladesh ICPC was also held in 2022.

This could not keep going because the ICPC will always be shifted by a whole year.

To solve this problem, Egypt is going to host two ICPCs at the same time. The 2022 ICPC and the 2023 ICPC will both take place in Egypt in 2023, and that will get the ICPC back on schedule.

Afterwards, all the upcoming contests will take place in the same year.

Given the year for which an ICPC contest was originally scheduled for, print "Egypt" if it is going to be hosted in Egypt. Otherwise, print "Elsewhere".

Input

The input contains a single integer Y $(1995 \leq Y \leq 2050)$ — the year for which an ICPC contest was originally scheduled.

Output

Print one string; "Egypt" if this ICPC be hosted in Egypt, or "Elsewhere" otherwise (without the quotes).

Examples

standard input	standard output
`1995`	`Elsewhere`
`2022`	`Egypt`
`2023`	`Egypt`
`2030`	`Elsewhere`

Problem J. Just A Different Order

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`White`

You have a list of N pairs. Each pair consists of a string (in the first position) and an integer (in the second position). Different pairs can also have similar first values, but will always have different integer values.

The list is sorted lexicographically by first values (strings). When two strings have the same value, the pairs are sorted ascendingly by the second value (integers). We want to rearrange the elements in the list in a specific manner.

First, we want to group all elements with the same first value together. Then, we will iterate through the pairs and pick only one pair from each group, the chosen pair should be the one with the current minimum second value, this pair should be added to the new list.

Then, we do the same operation on the remaining items over and over again, until no groups are left. When a group becomes empty, we ignore it in later iterations.

You should print the new list.

Input

The first line contains a single integer T — the number of test cases.

For each test case:

- The first line contains a single integer N $(1 \leq N \leq 10^5)$ — the number of pairs in the list.
- Each of the next N lines contains a space-separated pair $(S_i,\ I_i)$ $(1 \leq |S_i| \leq 50,\ 0 \leq I_i \leq 10^9)$ — the string and integer of the i^{th} pair, respectively.

  The N lines describing the pairs will be **sorted**, as described earlier.

The sum of N for all test cases will not exceed 10^5.

The characters given in all strings will be uppercase English letters.

Output

For each test case, print N lines. Each line should contain a space-separated pair of a string and an integer (in the desired order).

Example

standard input	standard output
1 8 A 1 A 2 A 3 B 5 B 7 C 6 C 8 D 4	A 1 B 5 C 6 D 4 A 2 B 7 C 8 A 3

Note

For example, the original list is:

$$original\_list = [\ \{``A", 1\}, \{``A", 2\}, \{``A", 3\}, \{``B", 5\}, \{``B", 7\}, \{``C", 6\}, \{``C", 8\}, \{``D", 4\}\ ]$$

After the first iteration:

$$original\_list = [\ \{``A", 2\}, \{``A", 3\}, \{``B", 7\}, \{``C", 8\}\ ]$$

$$new\_list = [\ \{``A", 1\}, \{``B", 5\}, \{``C", 6\}, \{``D", 4\}\ ]$$

After the second iteration:

$$original\_list = [\ \{``A", 3\}\ ]$$

$$new\_list = [\ \{``A", 1\}, \{``B", 5\}, \{``C", 6\}, \{``D", 4\}, \{``A", 2\}, \{``B", 7\}, \{``C", 8\}\ ]$$

After the third iteration:

$$original\_list = [\ ]$$

$$new\_list = [\ \{``A", 1\}, \{``B", 5\}, \{``C", 6\}, \{``D", 4\}, \{``A", 2\}, \{``B", 7\}, \{``C", 8\}, \{``A", 3\}\ ]$$

Now that the original list is empty, we print the new list.

Problem K. K Inversions

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Silver`

Given N and K, find the lexicographically smallest permutation of size N, that has K inversions.

An inversion in a permutation is a pair of indices such that the smaller index has the larger value. (i.e.) For a pair (P_i, P_j) to be inverted, the condition $i < j$ and $P_i > P_j$ holds.

Input

The first line of input contains T — the number of test cases.

For each test case:

- The input contains two space-separated integers N and K ($1 \le N \le 10^6$, $0 \le K \le \frac{N\times(N-1)}{2}$).

The sum of N for all test cases will not exceed 10^6.

Output

Print a space-separated list of integers — the answer permutation.

Example

standard input	standard output

5
5 3
7 2
8 4
6 3
6 15
	1 2 5 4 3
	1 2 3 4 6 7 5
	1 2 3 4 6 8 7 5
	1 2 3 6 5 4
	6 5 4 3 2 1

Note

For example $P = [2,\ 1,\ 4,\ 3,\ 5,\ 6]$ has two inversions of index pairs: (0, 1) and (2, 3).

Problem L. Looking for Numbers

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Rose`

There are N people, and M relations between them of the following form: person X knows the phone number of person Y (that doesn't necessarily mean that person Y knows the phone number of person X).

To reach someone, you can call them directly or call someone who can call them, or call someone who can call someone who can call them and so on...

The person with index i wants to be able to reach all friends with indices greater than i and does not care about the others.

What is the minimum number of times that each person needs to explicitly ask someone for their phone number, in order to be able to reach everyone greater than his index? In other words, what is the minimum number of edges that each person has to add to the graph, so they can reach everyone with an index greater than theirs?

Input

The first line contains two space-separated integers N and M ($2 \le N \le 10^6$, $0 \le M \le 10^6$) — the number of people and number of phone number relations.

Each of the next M lines contains a space-separated pair of indices $(X_i,\ Y_i)$ ($1 \le X_i < Y_i \le N$) — which means the person with index X_i has the phone number of person with index Y_i. **It is guaranteed that** $X_i < Y_i$.

Output

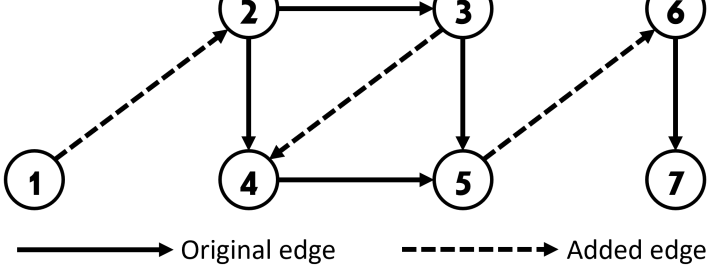
Print N space-separated integers on one line. The i^{th} integer corresponds to the minimum number of times person #i needs to ask someone for their phone number.

Example

standard input	standard output

7 5
2 4
2 3
4 5
6 7
3 5
	1 0 1 0 1 0 0

Note



In the example; persons #6 and #7 can already reach all people > their index.

Person #5 does not have the number of persons #6 and #7, but asks person #6 for their phone number, and reaches person #7 through them.

Person #4 can then reach persons #5, #6 and #7.

Person #3 can reach person #5. Thus, they reach persons #6 and #7 through person #5, and then ask person #4 for their phone number.

Person #2 can then reach persons #3, #4, #5, #6 and #7.

Person #1 is not connected to anyone, so asks person #2 for their phone number and can then reach everyone else through them.