Problem A. Saving the cinema

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

Those who are not familiar with movies often confuse the Star Wars and Star Trek sagas, which is offensive because Star Wars is widely superior. Esomer is tired of these offenses, so he wants to create a program that, given a character, responds whether it belongs to the Star Wars saga or the Star Trek saga. Unfortunately, he is not very good at programming, so he needs your help.

Input

The input will consist of a single string s, which will be one of the following three options (without quotes): “Yoda“, “Spock“, or “Frodo“.

Output

In the case that s is “Yoda“, you should print: “Pertenece a Star Wars.“.

In the case that s is “Spock“, you should print: “Pertenece a Star Trek.“.

In the case that s is “Frodo“, you should print: “No pertenece ni a Star Wars ni a Star Trek.“.

All of them without quotes.

Examples

standard input	standard output
`Spock`	`Pertenece a Star Trek.`
`Yoda`	`Pertenece a Star Wars.`
`Frodo`	`No pertenece ni a Star Wars ni a Star Trek.`

Note

—

Problem idea: Esomer

Problem preparation: Esomer

Occurrences: Novice 1, Advanced 1

Problem B. Operation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

José has just learned the basic arithmetic operations (addition, subtraction, multiplication, and division). Since he is quite good at it, his friend Óscar, who is playing a new video game he bought, asks José for help with his homework. The teacher gives him two positive integers a and b and asks him to apply one of the 4 mentioned operations to obtain the highest possible result. In other words, he is asked to find the maximum value of $a?b$, where ? can be addition, subtraction, multiplication, or division.

Input

You are given two integers a and b. ($1 \leq a, b \leq 100$).

Output

You have to print a number, the maximum value after applying an operation to a and b.

Examples

standard input	standard output
6 3	18
8 1	9

Note

—

Problem idea: Hectorungo18

Problem preparation: Hectorungo18

Ocurrences: Novice 2

Problem C. Maximum profit

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

Rafa and Jan are like joints, how good they are but how bad they are.

There is a bank with a stack of n checks, each having a specific face value in dollars.

The bank has a unique policy: they allow their customers to redeem a limited number of checks per day.

Your task is to help the bank's customers determine the maximum total value they can redeem in a **single** day.

Input

The first line contains two integers n ($1 \leq n \leq 10^4$) and k ($1 \leq k \leq n$), representing the number of checks and the maximum number of checks the bank allows to be redeemed in a day, respectively.

The second line contains n integers $c_1, c_2, ..., c_n$ ($1 \leq c_i \leq 10^4$), representing the face values of the n checks.

Output

Output a single integer, the **maximum** total value that can be redeemed in a single day by choosing the optimal combination of k checks.

Example

standard input	standard output
3 2 11 5 10	21

Note

The optimal strategy is to take the first and the third checks to get a sum of 21 dollars.

—

Problem idea: danx

Problem preparation: danx

Ocurrences: Novice 3

Problem D. jbum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	128 megabytes

Javier, the natty bodybuilder who bench presses 1073741824 kg, is training bench press.

In order to reach the desired weight for the lift, Javier will have to use the *duplicator*. If he puts x discs into the *duplicator*, after 1 minute he will have twice as many. To use the machine again, he has to remove all the discs and put in the desired number again. Since he's loaded up on creatine and pre-workout, this action takes no time at all. Initially, he only has one disc that his friend Bumstead gave him. It's also important to note that the bar he uses for the lift is magical and weighs nothing.

Javier needs n discs to do his exercise. You need to help him determine the minimum number of minutes it will take for him to reach the desired weight. If the minimum number of minutes is m, you also need to tell him how many discs he should put into the *duplicator* in the i-th minute, in order to reach the weight in the minimum possible time.

If there are multiple ways to put discs into the *duplicator* to reach the weight n in m minutes, you should print the lexicographically smallest sequence.

A sequence of numbers a is lexicographically smaller than another sequence b of the same length, if there exists an index i such that $a_i < b_i$ and for all j such that $1 \leq j < i$, it holds that $a_j = b_j$.

Input

The only line contains an integer n ($2 \leq n \leq 10^9$), the weight that Javier wants to lift.

Output

The first line of the output should contain an integer m, the minimum number of minutes Javier needs to perform his set with the desired weight.

The second line should contain m integers, $x_1, x_2, \ldots, x_m$, where x_i is the number of discs that Javier puts into the *duplicator* in the i-th time. Remember that if there are multiple possible ways to reach the weight in m minutes, you should print the sequence that is lexicographically smallest.

Examples

standard input	standard output
1024	10 1 2 4 8 16 32 64 128 256 512
33	6 1 1 2 4 8 16

Note

—

Problem idea: Hectorungo18

Problem preparation: Hectorungo18

Ocurrences: Novice 4

Problem E. Looking for palindromes

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

Even the smallest palindrome can change the course of the future

— Galadriel, or maybe Javier

A palindrome is a string of characters that reads the same from left to right as it does from right to left. Javier, being the absent-minded boy that he is, has been thinking that "palindromio" is the correct spelling for almost a year now! To do him a favor, let's create a new definition before he realizes that he's mistaken.

A "palindromio" is a string of digits from 0 to 9 that is exactly one position away from being a palindrome. 101 and 22 are palindromes, while 100 and 20 are "palindromios". Not satisfied with this, Javier now wants to know how many "palindromios" can be created using exactly n digits from 0 to 9. Help him satisfy his curiosity.

Input

The first line contains an integer t, $(1 \leq t \leq 100)$, the number of questions Javier will ask you. Each of his questions will consist of a single integer n, the length of the "palindromio", $(1 \leq n \leq 50000)$.

Output

For each question, you should print the number of "palindromios" of length n. Since this number can be very very large, Javier will be satisfied if you print it $\mod 10^9 + 7$. This operation is performed using % in C++ and Python. Note that $(a + b) \mod P$ is the same as $(a \mod P + b \mod P) \mod P$, and a similar thing applies with the product, so you can apply the modulo operation after every operation you make to avoid overflow. If you have questions about modulos, we will answer them in the clarifications.

Example

standard input	standard output
4 2 3 2023 13	90 900 616533557 540000000

Note

—

Problem idea: Javi

Problem preparation: Javi

Ocurrences: Novice 5

Problem F. Harry Potter in CMS

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Harry is participating in the Hogwarts Informatics Olympiad. There is 1 problem with many subtasks. However, Voldemort has cast a spell on the judges, making them believe that Harry has copied in some of his submissions. To find out how Harry is doing in the competition considering Voldemort's spell, Harry will ask you q queries.

The queries can be of 3 types:

- 1 k: Harry gives you information about a submission. He provides k numbers $x_1, x_2, \ldots, x_k$, where each x_i represents the index of a subtask that Harry has solved correctly in this submission. All the x_i are distinct. Note that Harry may solve a subtask correctly that he had already solved correctly in a previous submission.
- 2 i: Harry tells you that the judges have invalidated the submission he specified in query i. It is guaranteed that query i is of type 1 and that Harry had already made that query. It is also guaranteed that the submission had not been invalidated yet.
- 3: You must print the number of submissions that are currently increasing Harry's score (considering only the previous queries). A submission increases Harry's score if, for any of the subtasks he solves correctly, it is the first submission that Harry has specified to you and has not been invalidated with that correct subtask yet.

Input

The first line contains an integer t, the number of cases to answer. ($1 \leq t \leq 100$).

The first line of each case contains an integer q, the number of queries that Harry asks you. ($2 \leq q \leq 2 \cdot 10^5$).

The following q lines contain the queries. Line j contains an integer $type_j$, indicating the type of the query ($1 \leq type_j \leq 3$). If $type_j = 1$, it will be followed by an integer k_j and k_j integers $x_{j_1}, x_{j_2}, \ldots, x_{j_k}$, the subtasks that Harry solves in that submission, ($1 \leq x_{j_i} \leq 2 \cdot 10^5$). If $type_j = 2$, it will be followed by an integer i_j, the index of the query specifying the invalidated submission.

It is guaranteed that both the sum of q and the sum of $\sum k_j$ over all cases are at most $2 \cdot 10^5$.

Output

For each case, you must print the number of submissions that are increasing Harry's score for each query of type 3 that you receive.

Example

standard input	standard output
2 10 1 3 3 2 5 3 1 1 3 1 1 2 2 1 3 2 3 3 2 4 3 11 1 2 2 3 1 1 2 1 1 3 1 1 5 3 2 1 3 1 2 2 3 2 2 2 3 3	1 2 1 0 2 3 2

Note

—

Problem idea: Esomer

Problem preparation: Esomer, Hectorungo18

Ocurrences: Novice 6

Problem G. XOR + Constructive = Love

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

Litusiano has, yet again, found another 8-pages-long statement. Disappointed with the problemsetters who create such long statements, he has created a problem with a very short statement. Now he has gifted it to you so that you can enjoy a short statement, for once. Unfortunately, Esomer is very evil and he has intercepted Litusiano's gift, adding this whole useless paragraph to annoy Litusiano.

You are given an array a of length 30 and you need to construct a new array b of minimum length such that:

- There are at least a_i elements with bit i set in their binary representation, for $0 \le i < 30$.
- The sum of all the elements is s.
- The bitwise XOR† of all the elements is exactly x.

If such an array doesn't exist, output -1 instead.

†If you don't know what bitwise XOR is, remember that you can use the internet!.

Input

The first line of the input contains a single integer t, the number of test cases. $(1 \le t \le 10^4)$.

On the first line of each test case there are two integers, s and x, the required sum of all the elements in b and the required XOR of all the elements in b, respectively. $(0 \le s \le 10^{18}; 0 \le x < 2^{30})$.

Then comes a line with 30 integers, $a_1, a_2, \ldots, a_{30}$, each a_i denotes the minimum number all elements in b that must have the i-th bit set on their binary representation. $(0 \le a_i \le 10^9)$.

Output

For each test case, if there doesn't exist any array that satisfies the conditions, print -1. Otherwise, print the minimum length of such an array.

Examples

standard input	standard output

`2`
`9 5`
`1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0`
`9 6`
`1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0`
	`2`
	`-1`

`3`
`9 1`
`1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0`
`2147483648 1`
`1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0`
`104 10`
`5 2 3 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0`
	`2`
	`-1`
	`6`

Note

—

Problem idea: Esomer

Problem preparation: Esomer

Ocurrences: Novice 7, Advanced 2

Problem H. Menorca's ants

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

Dani is one of the best competitive programmers in the world, boyfriend of the jambilla, best friend of Innokentiy and bodybuilder.

One day in Menorca, he left some cookies at his bed and a lot of ants spawned there.

Hugo and Luis said him to trow the bed through the window and Dani did it, but Ilya and the funator were be very angry with them.

Now Dani needs help with the following problem:

There are n type of ants, and a_i ants of the i-th type, you have to find the **smallest** number of moves to throw **at least** p ants through the window.

Note that int each move Dani can throw **at most** m ants and k ants of the **same** type.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line consists in three integers n, m, k ($1 \le n \le 10^6; 1 \le m, k \le 10^{18}$) — the number of types of ants, the maximum number of ants that can be thrown on a move and the maximum number of ants of the same type that can be thrown on a move, respectively.

Then follows n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^{12}$) — the number of ants of each type.

The last line consist in an integer p ($1 \le p \le \sum_{i=1}^{n} a_i$) — the minimum number of ants that have to be thrown.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each testcase, output a single integer — the smallest number of moves.

Example

standard input	standard output
4 4 3 2 3 7 2 1 13 3 2 2 2 2 3 6 2 3 2 10 1 10 1 1 1 1 1	5 3 5 1

Note

The answer for the first testcase could be 1-st move: $(3, 2, 2)$, 2-nd move: $(1, 2, 2)$, 3-rd move: $(1, 2, 1)$, 4-th move: $(2, 3, 4)$, 5-th move: (2).

It can be proven that at least 5 moves are needed.

The answer for the second testcase could be 1-st move: $(1, 1)$, 2-nd move: $(2, 3)$, 3-rd move: $(3, 3)$.

It can be proven that at least 3 moves are needed.

—

Problem idea: danx

Problem preparation: danx

Ocurrences: Novice 8, Advanced 4

Problem I. Fake bills

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

The bank Monopoly has recently been very worried by the possibility of a robbery, as they don't want to lose their ~~fake~~ bills. To reinforce their security, they have implemented a system of security cameras in front of their vault. Now they are worried that it might have flaws, so they have asked for your help.

The room where the vault is located can be represented as an $n \times n$ grid, where the cell $(1, 1)$ is the entrance, and the vault is in the cell (n, n). There are m cameras in the room, each of them occupying exactly one cell. The cameras can point in 4 different directions: up, down, left and right. Each camera initially points in one direction, which may be different between cameras. When a camera points in one direction, it covers all the cells in that direction, including the cell in which it is located.

More formally, let the camera be in the cell (r, c), where r represents the row from the top and c the column from the left, then:

- If the camera points up, for all $1 \leq i \leq r$, the camera covers the cell (i, c).
- If the camera points down, for all $r \leq i \leq n$, the camera covers the cell (i, c).
- If the camera points left, for all $1 \leq j \leq c$, the camera covers the cell (r, j).
- If the camera points right, for all $c \leq j \leq n$, the camera covers the cell (r, j).

If a robber is at a cell covered by a camera, he will be caught.

Additionally, as Monopoly's owners are very paranoic, each second the cameras will rotate 90° **anticlockwise** (i.e. if a camera points down, it will point to the right; if it points to the right, it will point up; and so on). The robbers that try to break into the vault will have to start at cell $(1, 1)$ and reach the cell (n, n) without being seen by the cameras. Each second, robbers can do one of two things:

- Stay at the cell where they are.
- Move to an adjacent cell (i.e. a cell which shares a side with the cell they are in).

Note that **the cameras and the robbers move simultaneously**, so a robber can move to a cell which is currently covered by a camera as long as it is uncovered on the next second. See the samples section for an example of this.

Your task is to determine if a robber can reach the vault.

Input

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

The first line of each test case contains two integers n and m, denoting the size of the grid and the number of cameras, respectively. $(1 \leq n \leq 1000;\ 0 \leq m \leq \max(0, n^2 - 2))$.

Then follow m lines, each of them with two integers r and c, representing the row and column of the camera, and a character d which will be one of 'U', 'D', 'L', 'R', representing the initial direction of the camera. $(1 \leq r, c \leq n)$.

It is guaranteed that no two cameras share the same cell, that no cameras initially cover the cell $(1, 1)$, and that there isn't a camera at the cell (n, n).

Additionally, it is guaranteed that the sum of n over all test cases is at most 1000.

Output

For each test, you should output "YES" if a robber can reach the vault or "NO" if he can't.

Example

standard input	standard output
3 3 1 2 2 U 3 3 2 2 U 3 2 R 1 3 U 3 2 1 3 U 2 3 U	YES NO NO

Note

—

Problem idea: Esomer

Problem preparation: Esomer

Ocurrences: Novice 9

Problem J. Force Perturbation

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

The impacts of the creation of "palindromio" by Javier and his friends have reached Esomer (who is capable of measuring when the forces of order become unbalanced in the universe).

Esomer is able to sense that the Force has become unbalanced in an integer x. Being the busy sage that he is, he doesn't have time to restore it and will need to make use of his n students to restore the Force. Each student has a power of a_i, and choosing them will reduce the difference of x to $x - a_i$. Every day that passes, all his students increase their power by 1.

Help him remedy this perturbation as soon as possible, that is, print the minimum number of days in which the difference x can be reduced to **exactly** 0 using his students.

Once Esomer has used one of his students, due to their fatigue, he will not be able to use them again.

Input

The first line contains an integer t, $(1 \leq t \leq 10)$, the number of cases to solve.

Each of the cases will consist of two integers n, the number of students he has $(1 \leq n \leq 200)$, and x, the imbalance that has occurred in the force $(1 \leq x \leq 200)$.

n integers will follow, the sequence $a_1, a_2, ..., a_n$, where a_i represents the strength of the i-th student $(1 \leq a_i \leq x)$.

It is guaranteed that the sums of n and x over all cases will not exceed 200.

Output

For each case, you should print a single integer, the minimum number of days that must pass for Esomer to reduce x to 0.

Example

standard input	standard output
1 3 4 1 1 1	1

Note

—

Problem idea: Javi

Problem preparation: Javi

Ocurrences: Advanced 3

Problem K. Óscar and his battle

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes

Óscar is playing a video game. His goal is to collect as many coins as possible. To do this, he has to choose one of the n available characters. The i-th character has an attack power of a_i points and a defense power of b_i points. To collect the coins, he must defeat monsters (there are m monsters). The j-th monster has an attack power of c_j points, a defense power of d_j points, and rewards e_j coins when defeated.

In order to defeat a monster, Óscar needs his character's attack power to be **greater than or equal to** the monster's defense power, and his character's defense power to be **greater than or equal to** the monster's attack power. In other words, character i can defeat monster j if and only if $a_i \geq d_j$ and $b_i \geq c_j$.

Óscar can only choose one character, but he can defeat as many monsters as he wants. Once a monster is defeated, it cannot be defeated again. What is the maximum number of coins that Óscar can collect?

Input

The first line of the input contains an integer t, the number of test cases. ($1 \leq t \leq 100$).

The first line of each test case contains two integers n and m ($1 \leq n, m \leq 2 \cdot 10^5$).

The next n lines contain two integers each, a_i and b_i; the attack and defense power of each character, respectively. ($1 \leq a_i, b_i \leq 10^9$).

The next m lines contain three integers each, c_j, d_j, and e_j; the attack power, defense power, and coins rewarded by monster j. ($1 \leq c_j, d_j, e_j \leq 10^9$)

It is guaranteed that the sum of n and m over all test cases is at most $2 \cdot 10^5$.

Output

For each test case, you should print the maximum number of coins that Óscar can collect with a single character.

Examples

standard input	standard output
1 4 5 3 1 4 5 4 2 2 2 3 5 1 2 1 1 1 1 1 4 4 1 3 3 1	4
1 6 6 2 7 2 5 4 1 7 5 1 2 7 5 6 8 1 4 4 6 8 1 1 3 6 3 7 2 4 8 4 8	9

Note

—

Problem idea: Hectorungo18

Problem preparation: Esomer

Ocurrences: Novice 10, Advanced 5

Problem L. Random intervals

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Dani has two integers n and m and a button which does the following when pressed:

- Generates n **random** intervals of the form (l, r) where $1 \le l \le r \le m$.

Dani has pressed the button **two** times, but he has only given you the n randomly generated intervals from the first press of the button. Please, help Dani to know the probability of **not** having any pair of intersecting intervals.

A pair of intervals (i, j) don't intersect if and only if $r_i \le l_j$ or $r_j \le l_i$. Note that it's not guaranteed that after the first button press no pair of intervals intersect. For more information check the notes section below the samples.

It can be proven that the probability can be represented as an simple fraction P/Q where Q is coprime to 998244353 . Output the value of $P \cdot Q^{-1}$ modulo 998244353 .

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 69$). The description of the test cases follows.

The first line consists in two integers n, m ($1 \le n \le 269; 1 \le m \le 6.9 \cdot 10^6$) — the number of generated intervals after pressing the button and the upper bound of the intervals endpoints, respectively.

Then follows n lines, each one consisting in two integers l_i, r_i ($1 \le l_i \le r_i \le m$) — the endpoints of the generated intervals.

It is guaranteed that the sum of n over all test cases does not exceed 269.

Output

For each testcase, output a single integer — the probability of not having any pair of intersecting intervals modulo 998244353.

Example

standard input	standard output
8 1 3 2 2 3 4 3 3 1 3 3 3 2 4 1 1 3 4 4 10 5 7 2 2 3 4 9 9 2 15 1 3 2 5 2 69 8 11 9 10 1 10 1 1 2 11 3 3 3 3	831870295 726721889 259543532 10510086 0 0 1 501184665

Note

In the first test case, there are 6 possible distributions, but only 5 are valid. This is because if the generated interval was $(1, 3)$ the intervals $(2, 2)$ and $(1, 3)$ would intersect.

In the fourth test case, there are 100 possible distributions, but only 22 are valid.

In the fifth testcase, the answer is 0 because some of the intervals generated after the first press of the button intersect.

—

Problem idea: danx

Problem preparation: danx

Ocurrences: Advanced 6

Problem M. The battle of Helm's Deep

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

The world changes, and all that once was strong now proves unsure. How shall any tower withstand such numbers and such reckless hate?

— Théoden, King of Rohan

There are n towers at Helm's Deep, each of them has a power of a_i and a strength of b_i. Rohan's strategists determine that the orcs will arrive in q waves and that in the j-th wave x_j orcs will attack tower y_j. In that wave, tower y_j will take a damage of $max(0, x_j - a_{y_j} \cdot p_{y_j})$, where p_{y_j} is the number of soldiers in tower y_j. Once any tower i has taken an amount of damage that is at least b_i, it will be conquered. If some orcs attack an already conquered tower they will just go back to their lines confused (orcs aren't very intelligent). When a tower gets conquered, every soldier there will die. At the start of each wave (before the attack of that wave happens), the inner walls will get an amount of damage equal to the number of conquered towers at that moment. If the inner walls receive a lot of damage Rohan will fall and all hope for humans will be lost, so you want to minimize the damage received after the q waves.

Rohan counts with m soldiers and they ask you to help them locate the soldiers strategically. You must create the aforementioned array p of length n, which must satisfy the following conditions:

- For all i, $0 \leq p_i \leq m$.
- $\sum p_i \leq m$.

Let the minimum damage that the inner walls take over all possible arrays p be d. You must determine the value of d and the **lexicographically smallest**† array p, so that the damage the inner walls take is exactly d.

† An array a is lexicographically smaller than a different array b of the same length if, at the first position i where they differ, $a_i < b_i$.

Input

The first line of the input contains a single integer t, the number of test cases. ($1 \le t \le 100$).

The first line of each test case contains three integers n, m and q, the number of towers, soldiers and waves, respectively. ($1 \le n \le 1000$; $0 \le m \le 1000$; $1 \le q \le 50000$).

Each of the following n lines contains two integers, a_i and b_i, the power and the strength of the i-th tower. ($1 \le a_i, b_i \le 10^9$).

Then come q lines, each with two integers x_j and y_j, the number of orcs and the tower they attack in the j-th wave, respectively. ($1 \le x_j \le 10^9$; $1 \le y_j \le n$).

It is guaranteed that neither the sum of n nor the sum of m will exceed 1000 over all test cases, and that the sum of q won't exceed 50000 over all test cases.

Output

For each test case, on the first line output d, the minimum damage the inner walls can take.

Then, output n integers, the elements of the array p.

Example

standard input	standard output
2 5 15 7 2 3 4 5 3 3 4 1 1 1 3 1 4 5 1 3 3 2 100 1 2 3 9 4 5 9 4 1 1 1 1 1 1 1 1 1 1 3 5 4 5 5 5 6 5	2 1 0 1 0 4 0 0 0 0 0 5

Note

—

Problem idea: Carlos Villagordo Espinosa

Problem preparation: Carlos Villagordo Espinosa

Ocurrences: Advanced 7

Problem N. The Omer's orange tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Omer is a mathematician who loves programming and the orange trees, when the office hours end he loves playing with his beautiful orange tree.

This orange tree consists of n oranges numbered from 1 to n, connected by $n-1$ bidirectional branches such that **all** oranges are connected between them by some sequence of branches. Also this orange tree is **rooted** in the orange 1, and each orange has a weight w_i ($1 < w_i <= n$) where all w_i are different, note that w is a permutation of length n.

While playing with his orange tree, Omer asks himself some questions of the following form:

Given u, a, b calculate the value $\sum_{i=a}^{b} f(u,i)$ where $f(u,i) =$ number of oranges with weight divisible by i in the subtree rooted at orange u.

As he is very busy taking care of his children, he needs you to help him with his orange tree and to answer all his questions.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 1000$). The description of the test cases follows.

The first line consists in two integers n, q ($2 \le n, q \le 2 \cdot 10^5$) — the number of oranges and the number of questions, respectively.

The second line contains a permutation of length n, $w_1, w_2, ...w_n$ — the weights of the oranges.

Then follows $n-1$ lines, each one consisting in two integers u_i, v_i ($1 \le u_i, v_i \le n$) — meaning that there is a branch between the oranges u and v.

Finally follows q lines, each one consisting in three integers u_i, a_i, b_i ($1 \le u_i \le n; 1 \le a_i \le b_i \le n$) — the questions asked by Omer.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

It is guaranteed that the sum of q over all test cases does not exceed $2 \cdot 10^5$.

Output

For each testcase, output q integers — the answer to the questions.

Example

standard input	standard output
2 5 4 1 5 3 4 2 2 1 2 4 3 5 3 2 1 2 4 2 3 5 1 1 5 3 2 2 2 6 2 1 1 2 1 1 1 1 1 2 1 2 2 2 1 2 2 1 1 2 2 2	4 3 10 1 2 3 1 1 1 0

Note

—

Problem idea: danx

Problem preparation: danx

Ocurrences: Advanced 8

Problem O. Bea the maximizer

Input file: `standard input`
Output file: `standard output`
Time limit: 3 seconds
Memory limit: 256 megabytes

Bea likes big things

Everyone know it

Bea has two arrays a and b, both of n integers.

A permutation of length n is an array of length n where each number between 1 and n appears exactly once.

Your task is to find a permutation p of length n which **maximizes** the following value:

$(a_1 + b_{p_1})$ & $(a_2 + b_{p_2})$ & $\ldots$ & $(a_n + b_{p_n})$.

& is the bitwise AND operator, If you don't know how it works, remember that you can search for it on the internet ;)

As there can be a lot of solutions, Bea also wants to find one that **minimizes** the maximum distance from the position of a value on the chosen permutation to it's original position in the sorted permutation, in other words, Bea wants to find one that minimizes the following value: $\max_{1 \le i \le n}(|i - pos_i|)$ where pos_i is position of i in the chosen permutation and $|x|$ is the absolute value of x.

Please, help Bea :(

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

The first line consists in one integer n ($2 \le n \le 1500$) — the size of the arrays.

Then follows n integers a_i ($0 \le a_i \le 10^9$) — the values of a.

Finally follows n integers b_i ($0 \le b_i \le 10^9$) — the values of b.

It is guaranteed that the sum of n over all test cases does not exceed 1500.

Output

For each testcase, output two integers per line, the maximum possible value chosing the permutation optimally, and the minimum maximum distance from the position of a value on the chosen permutation to it's original position in the sorted permutation, so that the maximum value continues to be obtained.

Example

standard input	standard output
4 5 5 13 4 10 0 3 0 2 11 15 3 7 1 2 5 8 1 3 1 2 3 5 4 6 2 1000000000 1000000000 1000000000 1000000000	8 1 2 1 7 2 2000000000 0

Note

—

Problem idea: danx

Problem preparation: danx

Ocurrences: Advanced 9

Problem P. Ski resort

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Esomer loves skiing. However, he hates ski lifts because they are very long and always very cold. Determined not to suffer any longer, he has tried to come up with a route that minimizes the time spent on ski lifts. Unfortunately, he is not very good at interpreting the map of the pistes, so he has asked for your help.

The ski resort that Esomer visits can be represented as a directed acyclic graph with n nodes and m edges (the pistes). Additionally, there are k ski lifts that connect node a to node b. Interestingly, all ski lifts satisfy the condition that it is always possible to reach a from b without using any ski lift. It takes Esomer 1 minute to traverse a piste or use a ski lift.

Esomer is interested in how much time he needs to spend on ski lifts in order to spend a total of x minutes at the ski resort, without stopping at any node even for a second, while starting at node y. Esomer can finish his route at any node and leave for home as soon as he has spent the required x minutes.

Please help Esomer.

Input

The first line contains an integer t, the number of test cases. $(1 \le t \le 100)$.

For each test case, the first line contains four integers n, m and k. $(2 \le n \le 10^5; 1 \le m \le \min(10^5, \frac{n \cdot (n-1)}{2});$ $1 \le k \le 10^2)$.

Then, there are m lines, each containing two integers, u and v, indicating that there is an edge from u to v. $(1 \le u, v \le n)$.

It is guaranteed that the graph formed by the m edges is acyclic.

Next, there are k lines, each containing two integers, a and b, indicating that there is a ski lift from a to b. $(1 \le a, b \le n;\ a \ne b)$.

It is guaranteed that it is possible to reach a from b without using any ski lift.

Finally, there is a line containing two integers x and y, the number of minutes Esomer wants to spend at the ski resort, and the node at which he starts, respectively. $(1 \le x \le 10^9; 1 \le y \le n)$. It is guaranteed that Esomer can reach a ski lift from node y.

It is guaranteed that the sum of n and m over all test cases is at most 10^5, while the sum of k over all test cases is at most 10^2.

Output

For each test case, you should print an integer s, the minimum number of minutes Esomer needs to spend on ski lifts in order to spend a total of x minutes at the ski resort.

Examples

standard input	standard output
1 6 5 2 2 4 5 4 6 2 6 4 3 4 2 6 4 6 4 2	1
1 10 13 5 3 4 9 4 5 8 9 2 4 2 1 10 10 9 9 8 7 5 5 2 9 3 10 2 3 2 8 10 2 9 4 3 4 1 5 7 12 7	2

Note

—

Problem idea: Carlos Villagordo Espinosa

Problem preparation: Carlos Villagordo Espinosa

Ocurrences: Advanced 10