Problem A. String

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

There is a string of length n, $S[l..r]$ represents the string concatenated from the lth character to the rth character, and S_{len} is the length of the string($S[1..S_{len}]$ represents the whole S string).

We define F_G as the number of positive integers x that satisfy the following conditions:

1. $1 \leq x \leq G_{len}$

2. $G[1,x] = G[G_{len} - x + 1, G_{len}]$

3. The length of the common part of the intervals $[1,x]$ and $[G_{len} - x + 1, G_{len}]$ is greater than 0 and is divisible by k.

Now ask for the value of $\prod_{i=1}^{n}(F_{S[1..i]} + 1)$ modulo 998244353.

Input

The first line of input is a positive integer $T(T \leq 10)$ representing the number of data cases.

For each cases:

first line input a string S of lowercase letters, no longer than 10^6.

second line input a positive integer $k(1 \leq k \leq S_{len})$.

Output

For each cases, output a line with a positive integer representing the answer.

Example

standard input	standard output
1 abababac 2	24

Note

Note that the stack space of the judge system is a bit small, please pay attention to the reasonable allocation of memory.

Problem B. Dragon slayer

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

Long, long ago, the dragon captured the princess. In order to save the princess, the hero entered the dragon's lair.

The dragon's lair is a rectangle area of length n and width m. The lower left corner is $(0, 0)$ and the upper right corner is (n, m).

The position of the hero is $(x_s + 0.5, y_s + 0.5)$.

The position of the dragon is $(x_t + 0.5, y_t + 0.5)$.

There are some horizontal or vertical walls in the area. The hero can move in any direction within the area, but cannot pass through walls, including the ends of walls.

The hero wants to go where the dragon is, but may be blocked by walls.

Fortunately, heroes have access to special abilities, and each use of a special ability can make a wall disappear forever.

Since using special abilities requires a lot of physical strength, the hero wants to know how many times special abilities need to be used at least on the premise of being able to reach the position of the evil dragon?

Input

The first line contains an integer $T (T \le 10)$ —the number of test cases.

The first line of each test case contains 3 integers $n, m, K (1 \le n, m, K \le 15)$ —length and width of rectangular area, number of walls

The second line of each test case contains 4 integers $x_s, y_s, x_t, y_t (0 \le x_s, x_t < n, 0 \le y_s, y_t < m)$ — the position of the hero and the dragon.

The next K lines , each line contains 4 integers $x_1, y_1, x_2, y_2 (0 \le x_1, x_2 \le n, 0 \le y_1, y_2 \le m)$ — indicates the location of the two endpoints of the wall, ensuring that $x_1 = x_2$ or $y_1 = y_2$.

Output

For each test case, a line of output contains an integer representing at least the number of times the special ability was required.

Example

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

Problem C. Backpack

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

Alice has a backpack of capacity m that she now wants to fill with some items!

Alice has n items, each of which has a volume v_i and a value w_i.

Can a number of items be selected from n items such that the backpack is exactly full (ie the sum of the volumes equals the backpack capacity)? If so, what is the maximum XOR sum of the values of the items in the backpack when the backpack is full?

Input

The first line contains an integer $T(T \leq 10)$ —the number of test cases.

The first line of each test case contains 2 integers $n, m(1 \leq n, m < 2^{10})$ —the number of items, the capacity of the backpack.

The next n lines , each line contains 2 integers $v_i, w_i(1 \leq v_i, w_i < 2^{10})$ — the volume and value of the item.

Output

For each test case, output a single line, if the backpack cannot be filled, just output a line of 1 otherwise output the largest XOR sum.

Example

standard input	standard output
1 5 4 2 4 1 6 2 2 2 12 1 14	14

Problem D. Ball

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

Give a chessboard of $M * M$, with N point on it. You should calculate how many solutions there ar to select 3 points to make the median distance between the distance between the 3 points is a prime number?

the distance between $(x1, y1)$ and $(x2, y2)$ is $|x1 - x2| + |y1 - y2|$

Input

Each test contains multiple test cases. The first line contains the number of test cases $T(1 \leq T \leq 10)$. Description of the test cases follows.

The first line of each test case contains two integers N, M

The next N lines each line contains two integers x_i, y_i

It's guaranteed there are no $i, j(i \neq j)$ satisfies both $x_i = x_j$ and $y_i = y_j$

$1 \leq N \leq 2000, 1 \leq M \leq 10^5, 1 \leq x_i, y_i \leq M$

Output

For each test case, print one integer — the answer to the problem.

Example

standard input	standard output
2 3 3 1 1 2 2 3 3 3 3 1 1 2 1 3 2	1 1

Problem E. Grammar

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

Student Z took the course "Principles of Complier"this semester, and he was very interested in the grammar.

A grammar can be represented by $G = (V_T, V_N, S, P)$.

V_T represents the terminal symbol.

V_N represents a non-terminal symbol.

S represents the start symbol, which is the initial symbol of the derivation process , and can also be regarded as a special non-terminal symbol.

P represents the production. The production is divided into left and right sides connected by"$->$". It means that the left side of the production can be replaced by the right side. For example,the production which left is S and right is Aa is $S- > Aa$,means that the symbol string Aa can be used to replace the symbol string S.

A derivation is the process of replacing a non-terminal in a symbol string with the right side of the corresponding production.

The leftmost derivation result of a grammar starts from the start symbol. Each derivation will replace the leftmost non-terminal symbol of the symbol string with the symbol string on the right side of the corresponding production. After several derivations the resulting string of symbols.

For example, for the grammar "G[S]:S->aaBB B->bb":"a "b"are terminal symbol,"B"is non-terminal symbol,"S"is start symbol,"S->aaBB"and "B->bb"is production,then S->aaBB->aabbB->aabbbb is a leftmost derivation of this grammar,"aabbbb"is the result.

Student Z couldn't wait to write a grammar, but the grammar he wrote was relatively simple and met the following conditions:

1. Each terminal symbol of this grammar is a lowercase English letter, so there are only 26 terminal symbols in total.

2. There are a total of n non-terminal symbols, and the i non-terminal symbol is represented by "$[i]$".

3. There is only one symbol on the left of each production, and it is a non-terminal symbol.

4. A non-terminal symbol can only appear on the left of a production once, that is to say, there will be a total of n productions in the grammar.

5. The first character on the right side of each production must be a terminal symbol, that is, there will be no such situation as "$[1]- > [2]a$".

6. Since Student Z didn't think about which symbol to use as the start symbol, any non-terminal symbol may become the start symbol of the grammar.

However, because Student Z does not have a good grasp of the rules of grammar, there is a recursive definition in the grammar (that is, it derives itself from a non-terminal symbol), then it may not eventually be possible to generate a symbol string composed entirely of terminal symbols. Therefore, for each leftmost derivation, he will perform at most 100^{100} derivations to ensure that the first 100^{100} characters of the final symbol string are terminal symbols.

Student z doesn't care about the recursive definition, He just wants to know what is the yth terminal symbol in the final string. And he will reselects a non-terminal symbol as the start symbol each time.

Input

The first line of input is a positive integer $T(T \leq 10)$ representing the number of data cases.

For each test cases, the first line inputs two positive integers $n(1 \le n \le 1000)$, $q(1 \le q \le 1000)$ represents the number of non-terminal symbols and the number of queries.The data guarantees that the total length of n productions does not exceed $100,000$ characters.

The next n lines, each with a string describing a production, data guarantee that the ith production has and only has the ith non-terminal symbol on the left.

The next q line, each line has two positive integers $x, y(1 \le x \le n, 1 \le y \le 10^{18})$ represents a query, x represents the xth non-terminal symbol used as the start symbol,y represents the position of the terminal symbol of the query in this query.

Output

For each query, output a line representing the y-th terminal symbol, and output -1 if the total length of the symbol string is less than y.

Example

standard input	standard output
1 3 4 [1]->have[1]ac [2]->myname[3]id [3]->no 2 9 3 2 1 20 3 3	i o e -1

Problem F. Travel plan

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Bob lives on a magical land. There are n cities and m roads on the land. The length of the i-th road is w_i, and the length is an integer between 1 and L. Each road connects two cities. The land can be viewed as a graph of n points and m edges.

This land is magical because Bob was surprised to find that there are no **simple circuits** with an even total length in this land!

Bob likes to travel. If Bob takes a **simple path** from x to y $(x < y)$, the happiness value is the greatest common factor (gcd) of the lengths of all roads on the path.

simple path:A path is called a simple path if the vertices on the path do not repeat each other.

simple circuit:A circuit in which the vertices are not repeated except for the first and last vertices is called a simple circuit

Bob wants to count all possible travel paths.

Define $F(k)$ as the total number of travel paths with happiness value k, modulo 998244353.

Please find $F(1) \bigoplus F(2) \bigoplus F(3) \bigoplus ... \bigoplus F(L)$,where $\bigoplus$ represents XOR.

Input

The first line contains an integer $T(T \leq 500)$ —the number of test cases.

The first line of each test case contains 3 integers $n, m, L(1 \leq n, L \leq 100000, 1 \leq m \leq 200000)$ —number of cities, number of roads, length range of roads.

The next m lines , each line contains 3 integers $u_i, v_i, w_i(1 \leq u_i, v_i \leq n, 1 \leq w_i \leq L)$ —.represents a road of length w_i connecting u_i, v_i.

It is guaranteed that there are no double edges and self-loops.

$1 \leq \sum n, \sum L \leq 500000, 1 \leq \sum m \leq 1000000$

Output

For each test case, output a line containing an integer representing the answer.

Example

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

Problem G. Treasure

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

There are n islands in A, and m bridges(undirected edge) connect these n islands. From any island, you can reach any other island through bridges.

There are infinite treasures on each island, but each treasure on the same island has the same property and the same value.Each property of treasure will appear on up to 10 islands.

There is a warrior who is going to hunt for treasures, starting from the x island, reaching some islands(include x)through bridges and obtaining treasures, but treasures of each property can only be obtained once, that is, if the warrior has reached more than one islands with the same property treasures, he will only choose to collect one of the treasures in these islands.

There is a guard on each bridge. If you want to pass the bridge, you need to pass the test of the guard, that is to say, the combat power of the warrior cannot be less than that of the guard.

In this problem,two events will occur in the A country:

0 x y: Treasures on the x island have increased in value, which means all treasures on this island will increase in value by y.

1 x y: A warrior with combat power of y sets off from the x island to hunt for treasure. He wants you to figure out the most value he can get in total.

Input

The first line of input is a positive integer $T(T \leq 5)$ representing the number of data cases.

For each test case, the first line has three positive integers $n, m, q(1 \leq n, q \leq 100000, n-1 \leq m \leq 200000)$, representing the number of islands in A country, bridges and inquiries.

The next line inputs n positive integers, and the ith number $c_i(1 \leq c_i \leq n)$ represents the property of the treasure on the i-th island.

The next line inputs n positive integers, and the ith number $val_i(1 \leq val_i \leq 100000)$ represents the value of the treasure on the i-th island.

The next m line, each line of three positive integers $u, v, w(1 \leq u, v \leq n, u \neq v, 1 \leq w \leq 100000)$ represents a bridge between two islands, u, v represent the two endpoints of the bridge, w represents the combat power guarded on this bridge.

The next q line, each line of three integers $op, x, y(0 \leq op \leq 1, 1 \leq x \leq n, 1 \leq y \leq 100000)$ represents an operation:

If $op = 0$, representing an event, the value of the treasure on the x island increases by y.

If $op = 1$, it represents a query, asking how much value can be obtained by a warrior with combat strength of y from the x island.

Output

For each query, output a line with a positive integer representing the maximum value the warrior has earned.

Example

standard input	standard output
1 5 4 5 1 1 1 2 2 1 2 3 1 2 1 2 1 1 4 3 2 3 1 2 5 2 1 3 1 1 3 2 0 1 5 1 3 1 1 3 2	3 5 6 8

Problem H. Path

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

Given a graph with n vertices and m edges.Each vertex is numbered from 1 to n.

Each edge i has its cost w_i,some edges are common edges and some edges are special edges.

When you pass through a special edge, the next step after passing this edge,you can reach any vertex in the graph. if you goto the vertice which an original edge i can arrived from current vertice, the cost become $w_i - K(0 \le w_i - K)$(if you used edge i), otherwise the cost will become 0(every vertice except the vertice which original edge can arrived from current vertice)

original edge includes all common edges and special edges.

Now you start at S,You need to calculate the minimum cost from the starting vertex to each vertex(If there is a situation where you cannot reach, please output "-1 ")

Input

Each test contains multiple test cases. The first line contains the number of test cases T. Description of the test cases follows.

The first line of each test case contains four integers n, m, S, K

The next m lines each line contains four integers x, y, w, t represent an directed edge connect x and y with cost w,$t = 0$ represents it's a common edge,$t = 1$ represents it's a special edge.

$1 \le \sum m, \sum n \le 10^6, 1 \le x, y, S \le n, 1 \le w, K \le 10^9$

$K \le w_i (1 \le i \le m)$

Output

For each test case, print n integer in a line— the answer to the problem. There is a space at the end of the line for each line.when you cannot reach, please output -1.

Example

standard input	standard output
2 5 4 1 1 1 2 4 0 1 3 5 0 3 4 3 1 4 5 3 0 5 3 1 1 1 2 4 0 1 3 5 0 3 4 3 1	0 4 5 8 10 0 4 5 8 8

Problem I. Laser

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

There are n enemies on a two-dimensional plane, and the position of the i-th enemy is (x_i,y_i)

You now have a laser weapon, which you can place on any grid (x, y)(x, y are real numbers) , and the weapon fires a powerful laser that for any real number k, enemies at coordinates $(x + k, y), (x, y + k), (x + k, y + k), (x + k, y - k)$ will be destroyed.

You are now wondering if it is possible to destroy all enemies with only one laser weapon.

Input

The first line of input is a positive integer $T(T \leq 10^5)$ representing the number of data cases.

For each case, first line input a positive integer n to represent the position of the enemy.

Next n line, the i-th line inputs two positive integers $x_i, y_i(-10^8 \leq x_i, y_i \leq 10^8)$ represents the position of the i-th enemy.

The data guarantees that the sum of n for each test case does not exceed 500,000

Output

For each cases, If all enemies can be destroyed with one laser weapon, output "YES otherwise output "NO"(not include quotation marks).

Example

standard input	standard output
2 6 1 1 1 3 2 2 3 1 3 3 3 4 7 1 1 1 3 2 2 3 1 3 3 1 4 3 4	YES NO

Problem J. Walk

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

There is currently a grid of $n \times m$. You have to walk start at $(1, k_1)(\forall 1 \leq k_1 \leq m)$,end at $(n, k_2)(\forall 1 \leq k_2 \leq m)$.For every possible path, there will be a value V.The initial value of V is $f[k_1]$ when you start at $(1, k_1)$.When you reach (x, y), the value will become $V \times f[y]$.When you are located at (x, y) , you can walk to $(x + 1, P)(P \leq y + S(S(S(y))))$

Where $S(x) = \lfloor log2(max(1, x)) \rfloor$

Calculate the sum of the value of all the ways module 998244353.

Two ways A, B think different if $\exists(x, y)$, A passes (x, y) but B not.

Input

The first line contains two integers n, m

The second line contains m integers $f_1, f_2, ..., f_m$

$1 \leq n, m \leq 10^5, 0 \leq f_i \leq 10^9$

Output

print one integer — the answer to the problem.

Example

standard input	standard output
5 4 1 2 3 4	7770

Problem K. Random

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

N numbers, randomly generated between $[0, 1]$

Make M operation, $\frac{1}{2}$ probability to delete the maximum value, $\frac{1}{2}$ probability to delete the minimum value

Calculate the sum of expected value module $10^9 + 7$

Input

Each test contains multiple test cases. The first line contains the number of test cases $T (1 \le T \le 10000)$. Description of the test cases follows.

The first line of each test case contains two integers n, m

$1 \le m \le n \le 10^9$

Output

For each test case, print one integer — the answer to the problem.

Example

standard input	standard output
2 2 2 3 1	0 1

Problem L. Alice and Bob

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

Alice and Bob like playing games.

There are m numbers written on the blackboard, all of which are integers between 0 and n.

The rules of the game are as follows:

If there are still numbers on the blackboard, and there are no numbers with value 0 on the blackboard, Alice can divide the remaining numbers on the blackboard into two sets.

Bob chooses one of the sets and erases all numbers in that set. Then subtract all remaining numbers by one.

At any time, if there is a number with a value of 0 on the blackboard, Alice wins; otherwise, if all the numbers on the blackboard are erased, Bob wins.

Please determine who will win the game if both Alice and Bob play the game optimally.

Input

The first line contains an integer $T(T \leq 2000)$ —the number of test cases.

The first line of each test case contains a single integers $n(1 \leq \sum n \leq 10^6)$.

The second line of each test case contains $n+1$ integers $a_0, a_1, a_2...a_n (0 \leq a_i \leq 10^6, \sum a_i = m)$ — there are a_i numbers with value i on the blackboard .

Output

For each test case, print "Alice"if Alice will win the game, otherwise print "Bob".

Example

standard input	standard output
2 1 0 1 1 0 2	Bob Alice