Problem A. Bowling

Input file: `standard input`
Output file: `standard output`

Little Rabbit has recently become interested in a special kind of bowling. The bowling ball can be seen as a convex polygon on a two-dimensional plane. And the pins (the target of the bowling ball) can be seen as points on the plane.

As with regular bowling, the goal is to make the bowling ball hit as many pins as possible. We can suppose that the bowling ball makes a translational motion on the plane. Once the pin touches the bowling ball (including the boundary), the pin will be knocked down and will not affect the direction of the bowling ball's motion.

Now given the position of the bowling ball and the pins, for different directions of the bowling ball's motion, please calculate how many pins it can knock down.

Input

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

For each test case, the first line contains an integer n ($3 \le n \le 10^5$), indicating the number of vertices of the convex polygon.

Each of the next n lines contains two integers x, y ($|x|, |y| \le 10^9$), indicating that the coordinates of a vertex of the convex polygon are (x, y). The vertices are given in counterclockwise order, and there are no three vertices collinear.

The next line contains an integer m ($1 \le m \le 10^5$), indicating the number of pins.

Each of the next m lines contains two integers x, y ($|x|, |y| \le 10^9$), indicating that the coordinates of a pin are (x, y). It is guaranteed that the pins are located strictly at the outside of the polygon.

The next line contains an integer q ($1 \le q \le 10^5$), indicating the number of queries.

Each of the next q lines contains two integers x, y ($|x|, |y| \le 10^9$), indicating that the direction vector of the bowling ball's motion is (x, y). It's guaranteed that $(x, y) \ne (0, 0)$.

It is guaranteed that $\sum n$, $\sum m$, and $\sum q$ over all test cases do not exceed 2×10^5.

Output

For each query, output an integer in a single line indicating the number of pins the bowling ball can knock down.

Example

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

Problem B. Independent Feedback Vertex Set

Input file:	`standard input`
Output file:	`standard output`

Yukikaze loves graph theory, especially forests and independent sets.

- Forest: an undirected graph without cycles.
- Independent set: a set of vertices in a graph such that for every two vertices, there is no edge connecting the two.

Yukikaze has an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. Each vertex in V has a vertex weight. Now she wants to divide V into two complementary subsets V_I and V_F such that V_I is an independent set, and the induced subgraph $G[V_F]$ is a forest. The induced subgraph $G[V_F]$ is the graph whose vertex set is V_F and whose edge set consists of all of the edges in E that have both endpoints in V_F. In addition, she wants to maximize the sum of weights of vertices in V_I.

Since this problem is NP-hard for general graphs, she decides to solve a special case of the problem. We can build a special graph by the following steps. Initially, the graph consists of three vertices $1, 2, 3$ and three edges $(1, 2), (2, 3), (3, 1)$. When we add a vertex x into the graph, we select an edge (y, z) that already exists in the graph and connect (x, y) and (x, z). Keep doing this until there are n vertices in the graph.

Input

The first line of the input contains a single integer T ($1 \le T \le 10^3$), indicating the number of test cases.

The first line of each test case contains a single integer n ($4 \le n \le 10^5$), indicating the number of vertices in the graph. It is guaranteed that the sum of n over all test cases won't exceed 10^6.

The second line of each test case contains n positive integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$), indicating the weights of the vertices.

Initially, the graph consists of three vertices $1, 2, 3$ and three edges $(1, 2), (2, 3), (3, 1)$. The i-th line of the next $n - 3$ lines contains two integers u, v ($1 \le u, v < i + 3$), indicating the addition of a vertex $i + 3$ and two edges $(i + 3, u), (i + 3, v)$ to the graph. It is guaranteed that (u, v) already exists in the graph.

Output

For each test case, print an integer in a single line indicating the maximum sum of weights of vertices in V_I.

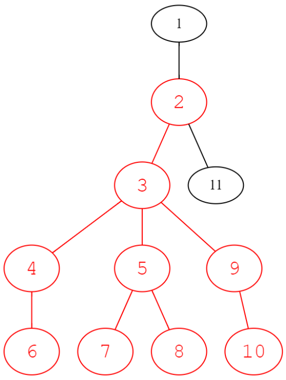
Example

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

Problem C. Counting Stickmen

Input file: `standard input`
Output file: `standard output`

Namesolo has fallen in love with the game *Stick Fight*. But when he is playing the game, he wonders how to find out the number of stickmen in the game.



In this question, we define the stick man as shown in the picture above. It has a chain of length 1 as the head, two chains of length 2 as the arms, a chain of length 1 as the body, and two chains of length 1 as the legs. For example, the red part in this picture can be viewed as a stick man, with chain $(2, 3)$ to be the head, chains $(3, 4, 6)$ and $(3, 9, 10)$ to be the arms, chain $(3, 5)$ to be the body, and chains $(5, 7)$ and $(5, 8)$ to be the legs.

The game can be viewed as a tree, and Namesolo wants to know how many stickmen are there. Please note that two stickmen are different when there is at least one different edge between the two edge sets that make up the two stickmen.

Because the answer may be too large, Namesolo wants to know the answer modulo 998244353.

Input

The first line of input contains one integer T $(1 \le T \le 15)$, indicating the number of test cases.

For each test case, the first line contains an integer n $(1 \le n \le 5 \times 10^5)$, indicating the number of vertices in the tree. Each of the following $n-1$ lines contains two integers a, b $(1 \le a, b \le n)$, indicating that there is an edge connecting a and b in the tree.

It is guaranteed that the sum of n over all cases won't exceed 3×10^6.

Output

For each test case, output an integer representing the answer modulo 998244353.

Example

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

Problem D. Black Magic

Input file: `standard input`
Output file: `standard output`

HoshiYo is learning *Black Magic* with n blocks. The left and right sides of each block are painted black or white. HoshiYo arranges the blocks in a row in a certain order without rotating them and then releases the *Black Magic*. Here's what happens next:

- For any two adjacent blocks, if the right side of the left block and the left side of the right block are both painted black, then the two sides will be pasted together making the two blocks into one.

HoshiYo wants to know the minimum and maximum blocks he can get after releasing the *Black Magic*.

Input

The first line contains an integer T ($1 \le T \le 4 \times 10^3$), indicating the number of test cases.

Each test case contains four integers E, L, R, B ($0 \le E, L, R, B \le 10^5, E + L + R + B \ge 1$), indicating the number of blocks.

- E: the number of blocks whose both sides are painted white.
- L: the number of blocks whose left side is painted black and right side is painted white.
- R: the number of blocks whose right side is painted black and left side is painted white.
- B: the number of blocks whose both sides are painted black.

It guaranteed that the sum of $E + L + R + B$ over all test cases won't exceed 10^6.

Output

For each test case, output two integers in a single line, indicating the minimum and maximum blocks HoshiYo can get.

Example

standard input	standard output
3 1 1 1 1 1 2 3 4 3 4 5 6	2 4 4 8 8 16

Note

Let's denote a block by (x, y), where x indicates the color on the left side, and y indicates the color on the right side. We use 0 to represent white and 1 to represent black.

For the first test case in the sample, here is a possible solution to get the minimum number of blocks:

$$(0,0) \quad (0,1) \quad (1,1) \quad (1,0)$$

As shown above, the last three blocks will be pasted into one.

And here is a possible solution to get the maximum number of blocks:

$$(0,0) \quad (1,0) \quad (1,1) \quad (0,1)$$

As shown above, any two blocks will not be pasted together.

Problem E. Cyber Painter

Input file:	`standard input`
Output file:	`standard output`

In the world of *Cyberpunk*, all paintings are done by using lasers. As a *cyber painter*, painting with lasers is your daily job.

You have a laser painting board with n rows and m columns of laser emitters. The distance between rows is 1, and so is the distance between columns. Each laser emitter can emit a laser with a length of 0.5 in four directions. Specifically, you can set an integer between 0 and 15 as the state value for each laser emitter, which can be denoted by a four-bit binary number $(X_1X_2X_3X_4)_2$ (For example, $11 = (1011)_2$). The meaning of the state value is as follows:

- $X_1 = 1$: The laser emitter emits a laser of length 0.5 in the upward direction.
- $X_2 = 1$: The laser emitter emits a laser of length 0.5 in the right direction.
- $X_3 = 1$: The laser emitter emits a laser of length 0.5 in the downward direction.
- $X_4 = 1$: The laser emitter emits a laser of length 0.5 in the left direction.

Given $n \times m$ integers between 0 and 15, you need to assign an integer to each laser emitter as its state value. You are curious about the expectation of the number of squares that can be formed by the laser if the $n \times m$ integers are assigned uniformly at random, where the squares can be of arbitrary edge length.

Input

The first line contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases.

The first line of each test case contains two integers n and m ($1 \le n \times m \le 10^5$), indicating the number of rows and columns of the laser emitters.

The second line of each test case contains 16 integers $a_0, a_1, \ldots, a_{15}$ ($0 \le a_i \le n \times m$, $\sum_{i=0}^{15} a_i = n \times m$), where a_i indicates the number of integer i.

It guaranteed that the sum of $n \times m$ over all test cases won't exceed 10^6.

Output

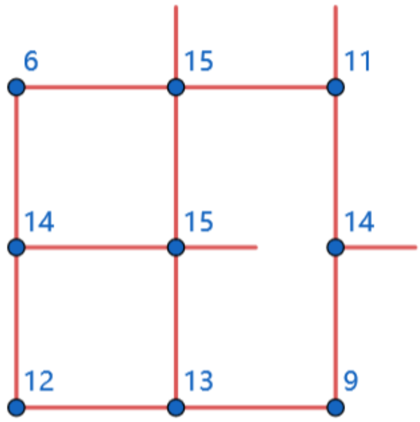
For each test case, output the expectation of the number of squares that can be formed by the laser in a single line. You should output the answer modulo 10^9+7. Formally, let $M = 10^9+7$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \times q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Example

standard input	standard output
3 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 2 2 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 3 3 0 0 0 0 0 0 1 0 0 1 0 1 1 1 2 2	1 41666667 41699736

Note

For the third test case in the sample, the following picture shows a possible assignment, which forms 3 squares.


6
15
11
14
15
14
12
13
9

Problem F. Sumire

Input file: `standard input`
Output file: `standard output`

Calculate

$$\sum_{i=l}^{r} f^k(i, B, d)$$

where $f(x, B, d)$ means the number of times that digit d appears in the base-B form of x (ignoring leading zeros).

In this problem, we consider that $0^0 = 0$.

Input

The first line contains one integer T ($1 \le T \le 10^4$), denoting the number of test cases.

For each test case, the only line contains five integers k, B, d, l, r ($0 \le k \le 10^9$, $2 \le B \le 10^9$, $0 \le d < B$, $1 \le l \le r \le 10^{18}$), as the statement shows.

Output

For each test case, output an integer indicating the answer modulo $10^9 + 7$ in a single line.

Example

standard input	standard output
3 2 2 0 1 5 1 4 3 11 45 10 14 11 19 198	6 19 1049

Note

For the first case in the sample, the answer is

$$\begin{aligned}&\sum_{i=1}^{5} f^2(i, 2, 0)\\&= 0^2 + 1^2 + 0^2 + 2^2 + 1^2\\&= 6\end{aligned}$$

Problem G. Weighted Beautiful Tree

Input file:	`standard input`
Output file:	`standard output`

A tree is a connected graph with n nodes and $n-1$ edges.

You are given a weighted tree with n nodes. The i-th node has a weight of wn_i and a cost of c_i. The i-th edge connecting node u_i and v_i has a weight of we_i. The edge is called beautiful if and only if it meets $\min(wn_{u_i}, wn_{v_i}) \le we_i \le \max(wn_{u_i}, wn_{v_i})$.

You can do the following operation several times.

- Choose a node u, then change its weight wn_u into wn'_u. The total cost adds $c_u\,|wn_u - wn'_u|$.

What is the minimum total cost to make all edges beautiful?

Input

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

For each test case, the input format is as follows:

$$\begin{array}{lllll} n \\ c_1 & c_2 & c_3 & \dots & c_n \\ wn_1 & wn_2 & wn_3 & \dots & wn_n \\ u_1 & v_1 & we_1 \\ u_2 & v_2 & we_2 \\ \vdots & \vdots & \vdots \\ u_{n-1} & v_{n-1} & we_{n-1} \end{array}$$

It is guaranteed that:

- $1 \le T \le 10^3$
- $1 \le n \le 10^5$, $\sum n \le 10^6$
- $1 \le c_i, wn_i, we_i \le 10^6$

Output

For each test case, output an integer in a single line, denoting the minimum total cost.

Example

standard input	standard output
2 3 2 1 2 9 9 10 1 2 10 1 3 11 3 1 1 2 9 9 10 1 2 10 1 3 11	3 2

Problem H. Triangle Game

Input file: `standard input`
Output file: `standard output`

Kate and Emilico are playing a game. There are 3 integers a, b, c. It is guaranteed that there exists a non-degenerate triangle whose side lengths are a, b, c respectively. The game goes as follows. Players take turns in decreasing a certain positive integer on one of the 3 integers. If there doesn't exist a non-degenerate triangle whose side lengths are a, b, c after a player's operation, the player loses.

Kate goes first. If both of them play optimally, will Kate win?

Input

The first line of input contains one integer T ($1 \le T \le 10^4$), indicating the number of test cases.

For each test case, the only line contains 3 integers a, b, c ($1 \le a, b, c \le 10^9$). It is guaranteed that there exists a non-degenerate triangle whose side lengths are a, b, c respectively.

Output

For each test case, if Kate will win, output `Win` in a single line. Otherwise, output `Lose` in a single line.

Example

standard input	standard output
3 2 2 3 2 3 4 5 3 4	Win Lose Win

Problem I. Counting Good Arrays

Input file:	`standard input`
Output file:	`standard output`

We consider an array consisting of positive integers $\{a_1, a_2, \ldots, a_n\}$ of length n is **good** if and only if for each $1 \le i < n$, a_{i+1} is divisible by a_i. Please note that we consider all the arrays with length 1 are good.

Given two integers n and m, please count the number of good arrays whose length is no greater than n and whose largest element is no greater than m. Since the answer may be large, you just need to output the answer modulo $10^9 + 7$.

Input

The first line of the input contains a single integer T $(1 \le T \le 10^3)$, denoting the number of test cases.

Each of the next T lines contains two integers n, m $(1 \le n, m \le 10^9)$, denoting a test case.

It's guaranteed that the number of test cases satisfying $\max(n, m) > 10^3$ will not exceed 50, the number of test cases satisfying $\max(n, m) > 10^6$ will not exceed 10, and the number of test cases satisfying $\max(n, m) > 10^8$ will not exceed 1.

Output

For each test case, output the answer modulo $10^9 + 7$ in a single line.

Example

standard input	standard output
5 2 4 3 5 10 12 24 17 114514 1919810	12 31 3915 190204 13530870

Note

All the good arrays with $n = 2, m = 4$ are:

- $\{1\}, \{2\}, \{3\}, \{4\}$
- $\{1, 1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}$
- $\{2, 2\}, \{2, 4\}$
- $\{3, 3\}$
- $\{4, 4\}$

Problem J. Connectivity of Erdős-Rényi Graph

Input file: `standard input`
Output file: `standard output`

Yukikaze is studying the theory of random graphs.

In the probability version of the Erdős-Rényi model, a random graph is constructed by connecting nodes randomly. That is, the random graph $G(n, p)$ is an undirected graph with n vertices, and each edge from the $\frac{n(n-1)}{2}$ possible edges is included in the graph with probability p independently from every other edge.

Now she wonders about the expected number of connected components in $G(n, p)$, modulo a large prime 998244353.

Input

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

The first line of each test case contains three integers q, a, b ($1 \le q \le 10^5$, $1 \le a \le b < 998244353$), denoting the number of queires and the probability $p = a/b$.

The second line of each test case contains q integers $n_1, n_2, \ldots, n_q$ ($1 \le n_i < 5 \times 10^5$ for each $1 \le i \le q$) seperated by spaces, denoting that Yukikaze wants to know the expected number of connected components in $G(n_i, p)$.

Let N be the sum of the maximum n_i of each test case, and Q be the sum of q of all test cases. It's guaranteed that $N \le 5 \times 10^5$ and $Q \le 10^5$.

Output

For each test case, output a single line containing the answers to the queries separated by spaces. You should output the answers modulo 998244353. That is, if the answer is $\frac{P}{Q}$, you should output $P \cdot Q^{-1} \bmod 998244353$, where Q^{-1} denotes the multiplicative inverse of Q modulo 998244353. We can prove that the answer can always be expressed in this form.

Don't output any extra spaces at the end of each line.

Example

standard input	standard output
3 1 14 51 4 1 91 98 10 2 114 514 1919 810	798850218 132789114 904977379 493892762

Problem K. Rock Tree

Input file: `standard input`
Output file: `standard output`

Professor Rockdu is interested in tree problems, and recently he has created a new data structure called *Rock Tree*.

Given a constant number k and a tree $T = \{V, E\}$ with V as the node set and E as the edge set, a non-empty set of nodes A is called a *Rock Tree* of T if and only if:

- $A \subseteq V$
- All nodes of A are connected in T, which means for every pair of nodes u and v which are both in A, the nodes in the shortest path between u and v in T are all in A.
- The largest distance over every two nodes in A is not greater than k. The distance between two nodes u and v is defined as the number of nodes (including u and v) in the shortest path between u and v in the tree.

Now Rockdu makes a tree R with n nodes and each node i has a value a_i assigned to it. He wants to find the *Rock Tree* with the maximum sum of node values.

Input

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

For each test case, the first line contains two integers n, k ($1 \le n \le 10^5$, $1 \le k \le n$), indicating the number of nodes and the distance limit.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($|a_i| \le 10^4$), indicating the value of nodes.

Each of the following $n - 1$ lines contains two integers u, v ($1 \le u, v \le n$), denoting an edge between u and v. It is guaranteed that these edges form a tree.

It is guaranteed that the sum of n over all test cases won't exceed 10^6, and there are at most 4 test cases with $n > 50000$.

Output

For each test case, output an integer denoting the maximum sum of node values in a single line.

Example

standard input	standard output
2 7 3 3 2 -2 -6 6 3 -7 1 2 2 3 2 4 2 5 5 6 2 7 12 5 0 7 -1 3 -3 10 -1 -1 -5 -1 -4 -9 1 2 1 3 1 4 2 5 4 6 6 7 1 8 8 9 5 10 9 11 9 12	11 20