Problem A. Typhoon

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	64 megabytes

A terrible typhoon is about to make landfall, and you decide to leave your home and seek refuge.

We can regard the problem as occurring on a two-dimensional plane. You have n alternative shelters, and you need to choose one to go to. Specifically, the i-th shelter is located at point (X_i, Y_i) .

Now the meteorological station gives the predicted typhoon track, which consists of m forecast points $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, where (x_1, y_1) is the starting point and (x_m, y_m) is the ending point. According to the forecast, the **center** of the typhoon will moves along the segments between two adjacent points in turn.

In order to simplify the model, it is assumed that the influence range of typhoon is a circle with radius r and the radius will never change during the move. If a point will be within or just on the boundary of the typhoon's influence range at a certain time, then we consider this point to be unsafe.

The radius of a typhoon is unpredictable. So you want to know, for each shelter, what is the minimum typhoon radius that still makes it unsafe.

Input

Please note that unlike other problems, there is only one test case input for this problem.

The first line contains two integers m, n ($2 \le m, n \le 10^4$), indicating the number of forecast points of the typhoon track and the number of shelters.

Then the following m lines, each line contains two integers x_i, y_i ($|x_i|, |y_i| \le 10^9$), indicating the coordinates of the i-th forecast points of the typhoon track.

Then the following n lines, each line contains two integers X_i, Y_i ($|X_i|, |Y_i| \le 10^9$), indicating the coordinates of the i-th shelter.

Output

Output n lines, the i-th line contains a single real number r_i , represents the answer for the i-th shelter.

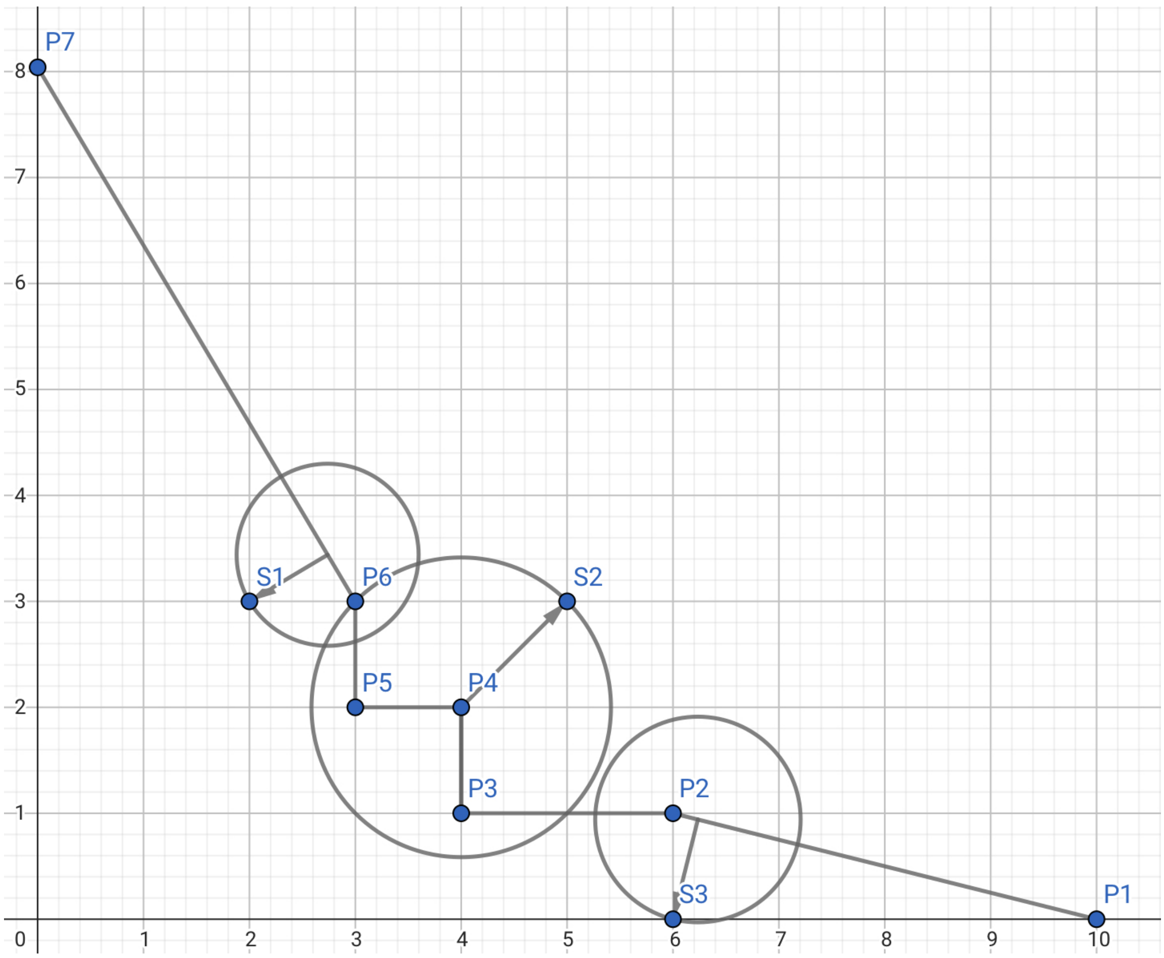
All the real number you output should be rounded into 4 decimals.

Example

standard input	standard output
7 3 10 0 6 1 4 1 4 2 3 2 3 3 0 8 2 3 5 3 6 0	0.8575 1.4142 0.9701

Note

The result of the sample is shown as the following image:


P7
S1
P6
S2
P5
P4
P3
P2
S3
P1
0
1
2
3
4
5
6
7
8
9
10

Problem B. GCD Magic

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	128 megabytes

Z is learning GCD theory and he finds a difficult problem:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}[\gcd(2^i-1,2^j-1)]^K$$

He doesn't know how to solve it, but he knows it's easy for you. Please help him.

Since the answer can be very large, you only need to print the answer mod 998244353.

Input

The first line contains one integer T $(1 \le T \le 12)$ which represents the number of test cases.

For each test case: One line contains two integers n $(1 \le n \le 10^9)$ and K $(0 \le K \le 10)$.

Output

For each test case: Print one line containing one integer which represents the answer.

Example

standard input	standard output
3 3 1 3 2 3 3	17 65 377

Problem C. String Magic (Easy Version)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

Z is learning string theory and he finds a difficult problem.

Given a string S of length n (indexed from 1 to n) , define $f(S)$ equal to the number of pair (i, j) that:

- $1 \le i < j \le n$
- $j - i + 1 = 2k, k > 0$ ($j - i + 1$ is even)
- $S[i, i + k - 1] = S[i + k, j]$
- $S[i, i + k - 1]$ is a palindrome

Here $S[L, R]$ denotes the substring of S with index from L to R.

A palindrome is a string that reads the same from left to right as from right to left.

To solve this problem, Z needs to calculate $f(S)$.

He doesn't know how to solve it, but he knows it's easy for you. Please help him.

Input

The first line contains one integer T ($1 \le T \le 10$) which represents the number of test cases.

For each test case: One line contains a string S ($1 \le |S| \le 10^5$).

It's guaranteed that the string only contains lowercase letters.

Output

For each test case: Print one line containing one integer which represents $f(S)$.

Example

standard input	standard output
3 aaaa abaaba ababa	4 2 0

Problem D. String Magic (Hard Version)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

Z is learning string theory and he finds a difficult problem.

Given a string S of length n (indexed from 1 to n) , define $f(S)$ equal to the number of pair (i, j) that:

- $1 \leq i < j \leq n$
- $j - i + 1 = 2k, k > 0$ ($j - i + 1$ is even)
- $S[i, i + k - 1] = S[i + k, j]$
- $S[i, i + k - 1]$ is a palindrome

Here $S[L, R]$ denotes the substring of S with index from L to R.

A palindrome is a string that reads the same from left to right as from right to left.

To solve this problem, Z needs to calculate $f(S[1, i])$ for each $1 \leq i \leq n$.

He doesn't know how to solve it, but he knows it's easy for you. Please help him.

Input

The first line contains one integer T $(1 \leq T \leq 10)$ which represents the number of test cases.

For each test case: One line contains a string S $(1 \leq |S| \leq 10^5)$.

It's guaranteed that the string only contains lowercase letters.

Output

For each test case: Print one line containing n integers, represents $f(S[1, i])$ for each $1 \leq i \leq n$.

Example

standard input	standard output
3 aaaa abaaba ababa	0 1 2 4 0 0 0 1 1 2 0 0 0 0 0

Problem E. Snake

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

There are n snakes indexed from 1 to n, each snake's length is 1 at first.

There will be battles between the snakes, each battle can take place between any two alive snakes.

After each battle, the winner eat the loser, and merge their body. For instance, if snake 1 battles with snake 2 and win, it will eat snake 2 and become $1 \to 2$, otherwise it will be eaten and the other snake will become $2 \to 1$. For another example, if snake $3 \to 2 \to 7$ battles with snake $5 \to 6 \to 4$, the winner will become $3 \to 2 \to 7 \to 5 \to 6 \to 4$ or $5 \to 6 \to 4 \to 3 \to 2 \to 7$.

In addition, if a snake's length is **greater than** k after a battle, it will become the king of the snakes!

A final state is defined by the set of alive snakes after battles. For instance, for $n = 3$, all possible final states are $\{1, 2, 3\}, \{1 \to 2, 3\}, \{1 \to 3, 2\}, \{2 \to 1, 3\}, \{2 \to 3, 1\}, \{3 \to 1, 2\}, \{3 \to 2, 1\}, \{1 \to 2 \to 3\}, \{1 \to 3 \to 2\}, \{2 \to 1 \to 3\}, \{2 \to 3 \to 1\}, \{3 \to 1 \to 2\}, \{3 \to 2 \to 1\}$.

Now you know after battles, there are m snakes alive and no king of the snake exist. Your task is to calculate the number of different possible final states modulo 998244353.

Input

The first line of the input contains an integer T ($1 \le T \le 200$), indicating the number of the test cases.

The next T lines, each line has three integers n, m, k ($1 \le m, k \le n \le 10^6$), indicating the number of snakes at first, the number of snakes after battles and the body length that need to be surpassed to become the king of the snake.

It's guaranteed that $\sum n \le 10^8$.

Output

T lines, each line has one interger, indicating the answer.

Example

standard input	standard output
7 3 3 3 3 2 3 3 1 3 6 2 3 6 2 4 6 2 5 1000000 114514 233	1 6 6 360 1080 1800 920789612

Note

Note that the answer could be zero!

Problem F. Touhou Red Red Blue

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

You are playing a game called Touhou Red Red Blue.

In this game, you will receive n mini UFOs (Undefined Fantastic Object) **one by one**, and you can decide to **store** or **discard** each UFO at the time you receive it.

Each UFO has one of these three colors: R, G, and B.

You have two bags numbered 1 and 2, which are initially empty.

When you decided to store a UFO (marked as U) and at least one bag is empty:

- If bag 1 is empty, store U in bag 1.
- If bag 1 is not empty and bag 2 is empty, store U in bag 2.

When you decided to store a UFO (marked as U) and no bag is empty, we consider the three UFOs we stored(the current one, the one in the bag 1 , and the one in the bag 2) :

- If all three UFOs have the same color, these three UFOs will disappear and you get 1 point (this is the only way you can get points). After that, you will get an additional UFO (**not in the given sequence**) in your bag 1 , which color can be decided by yourself.
- If all three UFOs have different colors with each other, these three UFOs will disappear. After that, you will get two additional UFOs (**not in the given sequence**) in your bag 1 and bag 2, and you can decide their colors independently by yourself.
- Otherwise, you will discard the UFO in bag 1, move the UFO in bag 2 to bag 1, and store U in bag 2.

What is the maximum points you can get after receiving all these n UFOs?

Input

The first line contains one integer T ($1 \leq T \leq 10$) which represents the number of test cases.

For each test case:

One line contains one string S ($1 \leq |S| \leq 10^5$) consisting of uppercase letters 'R', 'G' and 'B', represents the color sequence of all the given UFOs.

Output

For each test case, print one line containing one integer which represents the most points you could get.

Example

standard input	standard output
3 RRRRR RBGG RBRBR	2 1 1

Problem G. Expectation (Easy Version)

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

Note: The only difference between the easy and hard versions is the range of n and m .

You are to play a game for n times. Each time you have the probability $\frac{a}{b}$ to win.

If you win, you will get k^m score, where k is the total times you win at that time, otherwise you won't get any score.

Your final score is the sum of the score you get each time. For instance, if $n = 5, m = 10$, and you win twice in total, your final score is $1^{10} + 2^{10} = 1025$.

Now you wonder the expectation of your final score modulo 998244353.

Input

The first line of the input contains an integer T ($1 \le T \le 20$), indicating the number of the test cases.

The next T lines, each line has four integers n, m, a, b ($1 \le n \le 10^6, 1 \le m, a, b < 998244353$), indicating the number of games you play, the power of k is m, and the probatility to win a game is $\frac{a}{b}$.

It's guaranteed that $\sum n \le 10^7$.

Output

T lines, each line has one interger, indicating the answer.

Example

standard input	standard output
4 3 1 1 2 3 2 1 2 3 3 1 2 114514 1000000 123456789 987654321	748683267 4 748683273 733239168

Note

In first three test cases, you have the probability of $\frac{3}{8}$ to win once, $\frac{3}{8}$ to win twice, $\frac{1}{8}$ to win three times.

The expectation of your final score is $\frac{3}{8} \times 1^m + \frac{3}{8} \times (1^m + 2^m) + \frac{1}{8} \times (1^m + 2^m + 3^m)$.

So your first three answers are $\frac{9}{4}, 4, \frac{33}{4}$.

Problem H. Expectation (Hard Version)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	512 megabytes

Note: The only difference between the easy and hard versions is the range of n and m . Please refer to the time limit set on the online judge!!!

You are to play a game for n times. Each time you have the probability $\frac{a}{b}$ to win.

If you win, you will get k^m score, where k is the total times you win at that time, otherwise you won't get any score.

Your final score is the sum of the score you get each time. For instance, if $n = 5, m = 10$, and you win twice in total, your final score is $1^{10} + 2^{10} = 1025$.

Now you wonder the expectation of your final score modulo 998244353.

Input

The first line of the input contains an integer T $(1 \le T \le 20)$, indicating the number of the test cases.

The next T lines, each line has four integers n, m, a, b $(1 \le m \le 10^6, 1 \le n, a, b < 998244353)$, indicating the number of games you play, the power of k is m, and the probatility to win a game is $\frac{a}{b}$.

It's guaranteed that $\sum m \le 10^7$.

Output

T lines, each line has one interger, indicating the answer.

Example

standard input	standard output
4	748683267
3 1 1 2	4
3 2 1 2	748683273
3 3 1 2	733239168
114514 1000000 123456789 987654321	

Note

In first three test cases, you have the probability of $\frac{3}{8}$ to win once, $\frac{3}{8}$ to win twice, $\frac{1}{8}$ to win three times.

The expectation of your final score is $\frac{3}{8} \times 1^m + \frac{3}{8} \times (1^m + 2^m) + \frac{1}{8} \times (1^m + 2^m + 3^m)$.

So your first three answers are $\frac{9}{4}, 4, \frac{33}{4}$.

Problem I. Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	512 megabytes

There is a rooted tree of n vertices with the root in vertex 1, each vertex has zero or one **heavy child** linked with a **heavy edge**, a set of maximal connected heavy edges form a **heavy chain**. A single vertex can also form a heavy chain, so each vertex belongs to exactly one heavy chain.

Now we want to build a search tree to maintain some information of the original tree. For each heavy chain with k vertices, we construct a segment tree of depth $\lceil \log_2 2k \rceil$ to maintian it, with each leaf of the segment tree indicating a vertex on the heavy chain, and the parent of the segment tree's root is the parent of the top vertex of the heavy chian.

Note that in our variant of the segment tree, every leaves have the same depth.

For instance, this is a possible original tree, note that a double slash indicating a heavy edge.

```
          1
        // \
        2   7
      //    \\
      3       8
    //        \\
    4           9
  // \
  5  10
 /
6
```

There are 4 heavy chain in this tree. They are: $1 \to 2 \to 3 \to 4 \to 5$, 6 , $7 \to 8 \to 9$, 10

This is the search tree of the previous origin tree, note that the double slash here has no special meaning, just make the search tree looks more beautiful.

```
            o
          // \\
         o     o
       // \\   //
      o    o   o
     / \  / \ /
    1  2 3  4 5
    |       | |
    o      10 6
   / \
  o   o
 / \  /
7   8 9
```

Given such a original tree, your task is to calculate the depth of the search tree.

The depth of the root is 1.

Input

The first line of the input contains an integer T $(1 \leq T \leq 20)$, indicating the number of the test cases.

For each test case:

The first line of the input contains an interger n $(1 \leq n \leq 10^6)$, indicating the number of the vertices in the original tree.

The next lines has n integers, indicating the parent of each vertex. Note that 0 indicating don't have a parent.

The next lines has n integers, indicating the heavy child of each vertex. Note that 0 indicating don't have a heavy child.

It's guaranteed that $\sum n \leq 10^7$.

Output

T lines, each line has one interger, indicating the answer.

Example

standard input	standard output
1 10 0 1 2 3 4 5 1 7 8 4 2 3 4 5 0 0 8 9 0 0	7

Note

Note that the input scale is **extremely** large.

Problem J. Cut The Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	7 seconds
Memory limit:	128 megabytes

Alice and Bob are playing the game "Cut The Tree"on a tree T with n nodes indexed from 1 to n.

For each node u , there is a mysterious weights w_u on it.

The game will be played in the following order:

1. Alice choose an edge of the tree and cut it. So the tree T will be divided into two new trees T_1, T_2 .

2. Bob selects one of the two trees to take away, and gives the other to Alice.

3. Bob selects two nodes u, v from his tree ($u = v$ is allowed), and his score is $w_u \oplus w_v$, where $\oplus$ means bitwise XOR. And so for Alice, she selects two nodes u', v' from her own tree ($u' = v'$ is allowed), and her score is $w_{u'} \oplus w_{v'}$.

The final score of the game is Bob's score minus Alice's score. Bob wants to maximize it, while Alice wants to minimize it.

If both of them is clever enough, what's the final score of the game?

Input

The first line of the input contains an integer T $(1 \le T \le 5)$, indicating the number of the test cases.

For each test case:

The first line contains a single integer n $(2 \le n \le 2 \times 10^5)$, indicating the number of the nodes of the tree.

The second line contains n integers $w_1, w_2 \ldots, w_n$ $(0 \le w_i \le 10^9)$, indicating the weight of the nodes.

Then the following $n - 1$ lines, each line contains two integers u_i, v_i $(1 \le u_i, v_i \le n, u_i \ne v_i)$, indicating an undirected edge of the tree.

It's guaranteed that the graph is a tree.

Output

For each test case, print one line contains a single integer, indicating the final score of the game.

Example

standard input	standard output
2 5 1 2 3 4 5 1 2 2 3 3 4 4 5 9 4 52 32 34 65 57 26 18 12 2 1 3 1 4 3 5 1 6 5 7 4 8 7 9 7	2 62

Note

For the first sample, Alice will choose to cut the edge connected 3 and 4, then Bob's score will be 3 and Alice's score will be 1.

Problem K. Cactus Circuit

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	128 megabytes

An undirected connected graph is called a cactus, if and only if each edge of it belongs to at most one simple cycle. A simple cycle is a cycle that has no repeated vertices. Note that there is no self loop in the graph, but there may be duplicate edges connect same nodes.

There is an old circuit network contains n nodes and m undirected edges, and the network is guaranteed to be a cactus. Each edge (u_i, v_i) has a durability d_i , represents the length of time it can transport electricity.

Now your job is to choose an activation time s_i ($s_i \geq 0$) for each edge (u_i, v_i), then this edge will only transport electricity in the time interval $[s_i, s_i + d_i]$.

However, this fragile network cannot be disconnected from any node. Firstly, you must ensure that, the edges that can transport electricity at time 0 have connected all the nodes together. After that, once two nodes cannot be connected at a certain moment, the entire power network will immediately collapse.

The question now is, how long can you keep this network running?

Formally, if your answer is A , then you should have a way to set all s_i satisfies that, for every real number $t \in [0, A]$, all the n nodes are connected by the edges that can transport electricity at time t .

Input

The first line of the input contains an integer T ($1 \leq T \leq 5$), indicating the number of the test cases.

For each test case:

The first line contains two integers n, m ($1 \leq n \leq 10^5, 1 \leq m \leq 2 \times 10^5$), indicating the number of the nodes and the edges in the circuit network.

Then the following m lines, each line contains three integers u_i, v_i, d_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq d_i \leq 10^9$) , indicating an edge in the given graph. It's guaranteed that the graph is a cactus.

Output

For each test case: print one line contains a single integer, indicating the maximum length of time that you can keep this network running.

Example

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

Note

For the first sample, you can set $s_1 = s_3 = 0, s_2 = 1$ so all the nodes are connected by edge 1 and edge 3 in time interval $[0, 1]$, and by edge 2 and edge 3 in time interval $[1, 2]$.

Problem L. Counting Stars

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	64 megabytes

We define a k-star graph ($k \geq 2$) is a connected undirected graph with $k+1$ nodes and k edges satisfies that all the k edges are connected to a common node (and it's easy to show that this node is unique), and we call this node as the center of the star graph, and the other nodes are the leaves of the star graph.

Now you are given a simple undirected graph with n nodes and m edges.

Your task is to calculate how many k-star graphs are there in the given graph for all the integers $k \in [2, n-1]$.

Specifically, the problem can be described as follows:

How many ways are there to choose a node c and a set of k nodes $\{l_1, l_2, \ldots, l_k\}$ in the given graph, satisfies that there exists edges connected c and each l_i for all $i \in [1, k]$ in the given graph. Two star graphs are considered different if the center node c is different or the set of leaves $\{l_1, l_2, \ldots, l_k\}$ is different.

Let cnt_k represent the number of k-star graphs in the given graph **module** $10^9 + 7$.

You only need to print the bitwise XOR sum of all the cnt_k (i.e. $cnt_2 \oplus cnt_3 \oplus \cdots \oplus cnt_{n-1}$,where $\oplus$ represents bitwise XOR).

Input

The first line of the input contains an integer T ($1 \leq T \leq 5$), indicating the number of the test cases.

For each test case:

The first line contains two integers n, m ($3 \leq n \leq 10^6, 1 \leq m \leq 10^6$), indicating the number of nodes and edges in the given graph.

Then the following m lines, each line contains two integers u, v ($1 \leq u, v \leq n, u \neq v$) , indicating an edge in the given graph. It's guaranteed that the graph is simple.

Output

For each test case: print one line contains a single integer, indicating the bitwise XOR sum of all the cnt_k.

Example

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

Note

For the second sample, there are twelve 2-star graphs and four 3-star graphs, so the answer is $12 \oplus 4 = 8$.
Note that the input scale is **very** large.