1001.Multiply 2 Divide 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	7.5 seconds
Memory limit:	512 megabytes

Note:There is no dependency between this problem and problem Hack of Multiply 2 Divide 2.

Frank_DD has a sequence a of length n.

For each operation, he selects a number $a_i (1 \le i \le n)$ and changes it to $a_i \cdot 2$ or $\lfloor \frac{a_i}{2} \rfloor$.

Frank_DD wants to know the minimum number of operations to change the sequence a to a non-descending sequence.

Input

The first line of the input contains one integer T $(1 \le T \le 5\ )$ — the number of test cases. Then T test cases follow.

In each test case:

The first line contains a single integer $n (1 \le n \le 10^5)$ — the length of sequence a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^5)$ — the sequence a.

Output

For each test case, print a single integer in a single line — the minimum number of operations to change the sequence a to a non-descending sequence.

Example

standard input	standard output

2
7
6 3 3 4 10 8 2
10
9 9 4 7 3 10 10 8 4 3
	4
	11

Note

In the first test case, we can use at least 4 operations to change the sequence a to a non-descending sequence:

$a_1 = \lfloor \frac{a_1}{2} \rfloor$

$a_5 = \lfloor \frac{a_5}{2} \rfloor$

$a_7 = a_7 \cdot 2$

$a_7 = a_7 \cdot 2$

1002.Hack of Multiply 2 Divide 2

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

Note:There is no dependency between this problem and problem Multiply 2 Divide 2.

Frank_DD has a sequence a of length $n(1 \le n \le 10^5, 1 \le a_i \le 10^5)$.

For each operation, he selects a number $a_i(1 \le i \le n)$ and changes it to $a_i \cdot 2$ or $\lfloor \frac{a_i}{2} \rfloor$.

Frank_DD wants to know the minimum number of operations to change the sequence a to a non-descending sequence.

To help Frank_DD solve this problem, ddy guesses that the number in the result sequence can't be very large, and it will not exceed $2^{127} - 1$, which means that every number in the result sequence can be held in a 128-bit signed integer variable.

Formally, he guesses that for each sequence a that satisfies the constraints, there will always be a way that the number of operations is minimal, and in the result sequence, $a_n < 2^{127}$.

So, he writes a program relying on this idea.

But unfortunately, the idea is wrong, so this program can be hacked.

Now you should construct a sequence to hack the program of ddy. Your solution is considered correct if and only if this sequence satisfies the constraints, and the idea of ddy is wrong to this sequence.

Input

This problem has no input.

Output

The first line contains a single integer $n(1 \le n \le 10^5)$ — the length of sequence a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^5)$ — the sequence a.

Note

For example, if your output is

7

6 3 3 4 10 8 2

We can use at least 4 operations to change the sequence a to a non-descending sequence:

$a_1 = \lfloor \frac{a_1}{2} \rfloor$

$a_5 = \lfloor \frac{a_5}{2} \rfloor$

$a_7 = a_7 \cdot 2$

$a_7 = a_7 \cdot 2$

In this way, in the result sequence, $a_n = 8 < 2^{127}$. So you will get wrong answer.

1003.Find the Number of Paths

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	512 megabytes

Huah has $n+k$ cities numbered $1, 2, ..., n+k$, the city $i(1 \le i < n+k)$ to the city $i+1$ has $n+k-i$ distinct one-way roads.

For each $x = 1, 2, ..., n-1$,the city $i(x < i \le n+k)$ to the city $i-x$ has a_x distinct one-way roads.

For $m = k+1, k+2, ..., k+n$,find the number of paths from city $k+1$ to city m that pass through exactly k number of roads.

Two paths are distinct when and only if the sequence of edges they pass through is distinct and the answer is modulo 998244353.

Input

First line has one integer $T(1 \le T \le 14)$, indicating there are T test cases. In each case:

First line input two integers $n, k(2 \le n \le 2 \times 10^5, 1 \le k \le 2 \times 10^5)$.

Second line $n-1$ integers $a_1, a_2, ..., a_{n-1}(0 \le a_i \le 998244352)$.

There is a blank line between case $i(1 \le i < T)$ and case $i+1$.

Input guarantee $\sum(n+k) \le 1006769$.

Output

In each case, output a row of n integers with the i-th integer being the answer when $m = k+i$.

Example

standard input	standard output

4
3 2
1 2
 	
3 1
1 2
 	
5 10
2 3 3 3
 	
3 3
166374059 748683265
	5 0 2
	0 2 0
	114307026 825469567 425461680 73846080 5140800
	5 2 0

1004.Yet Another Easy Permutation Count Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 5 seconds
Memory limit:	256 megabytes

Silver187 likes Permutation. For a permutation P of length n, a position $x(2 \le x \le n-1)$ is a good position if and only if $\forall 1 \le i \le x-1, P_i < P_x$, and $P_x > P_{x+1}$. In particular:

1. position 1 is a good position if and only if $P_1 > P_2$ and $n \ge 2$.

2. position n can never be a good position.

Silver187 wants to calculate the beauty value of a permutation P of length n. He defines a number S, initially $S = 0$. Silver187 will repeat the following operations for the permutation P until the permutation P is in ascending order.

1. Add to S the number of good positions in the current permutaion P.

2. Do a bubble sort on the permutation P(For each i from 1 to $n-1$ in order, if $P_i > P_{i+1}$, swap P_i, P_{i+1}).

S is the beautiful value of the permutation P.

Silver187 gives you two numbers n and m. There are m constraints. Every constraint will give x and y, which means the inital number of position x is y. Find the sum of the beauty values of all permutations that satisfy all constraints modulo 998244353.

Input

The first line has one integer $T(1 \le T \le 100)$, indicating there are T test cases.

In each case:

The first line contains two integers $n(1 \le n \le 10^6)$, $m(0 \le m \le n)$—the length of the permutation and the number of constraints.

The i-th line of the next m line contains two integers—the i-th constraint.

It is guaranteed that there is at least one permutation that satisfies all constraints.

Input guarantee $1 \le \sum m \le \sum n \le 10^7$.

Output

In each case, output a single integer—the sum of the beautiful values of all permutations that satisfy the constraints modulo 998244353.

Example

standard input	standard output

2
3 1
1 2
7 5
4 5
2 2
6 7
3 3
1 4
	3
	13

1005.Yet Another Easy Function Sum Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 0 seconds
Memory limit:	5 1 2 megabytes

Two years ago, Silver187 learned Mobius inversion and knew how to calculate $(1 \le n \le 10^9)$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\gcd(i,j)$$

One year ago, Silver187 learned how to calculate $(1 \le n \le 10^5)$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi(ij)$$

But he tried to solve this problem when $1 \le n \le 10^9$. Finally, he failed to solve it. But he didn't completely fail, he solved a similar problem:

Silver187 defines that if $n = \prod_{i=1}^{k} p_i^{\alpha_i}(p_i \in \text{prime}, \alpha_i > 0, \forall i \ne j, p_i \ne p_j)$, then $H(n) = \prod_{i=1}^{k} p_i$.

Silver187 likes gcd, so he wants to ask you to calculate the result of the following formula.

$$(\sum_{i=1}^{n}\sum_{j=1}^{n} H(ij)[\gcd(i,j)=1]) \bmod 10^9+7$$

Now, Silver187 asks you to solve this problem.

Input

First line has one integer $T(1 \le T \le 5)$, indicating there are T test cases. In each case:

Only one line contains an integer $n(1 \le n \le 10^9)$.

Input guarantee $\sum n \le 2 \times 10^9$.

Output

In each case, output an integer on a line.

Example

standard input	standard output
5 3 5 1000 10000 1000000	23 119 181591410 452132610 74649566

1006.Maex

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	128 megabytes

You are given a rooted tree consisting of n vertices numbered from 1 to n, and the root is vertex 1.

Vertex i has a natural number weight a_i, and **no two different vertexes have the same weight**.

Define $b_u = MEX\{x|\exists v \in subtree\,(u)\,, x = a_v\}$.

Unfortunately, a_i are not given. Please find out the maximum possible $\sum_{i=1}^{n} b_i$.

The **MEX** of a set is the minimum non-negative integer that doesn't belong to the set.

Input

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

For each test case:

The first line contains one integer $n\,\left(1 \le n \le 5 \cdot 10^5\right)$, indicating the number of nodes.

In the following $n-1$ lines, each line contains two interger $u, v\,(1 \le u, v \le n)$, indicating an edge (u, v) of the tree.

A guarantee is that forming trees.

Output

For each test case: One line with an integer, indicating the maximum possible $\sum_{i=1}^{n} b_i$.

Example

standard input	standard output

3
5
1 2
3 2
1 5
4 1
3
1 2
2 3
1
	8
	6
	1

1007.Shinobu loves trip

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

As a cold-blooded, hot-blooded, and iron-blooded vampire, Shinobu loves traveling around.

There are P countries in total, numbered $0, 1, \ldots, P-1$.(It is guaranteed that P is a prime number)

It is known that when Shinobu is in the country numbered i, the next country she visits must be the country numbered $(i \cdot a)\%P$ (a is a constant parameter), and it takes Shinobu 1 day to go from one country to another.

In order to travel smoothly, Shinobu has customized n travel plans, and the i-th travel plan is represented by the starting country s_i and the travel days d_i.

For example, if $P = 233,\ a = 2$, a plan's starting country is 1 and travel days is 2, then Shinobu will visit the city $\{1, 2, 4\}$ according to this plan.

Playf knows these travel plans and the value of parameter a, now he wants to ask you q questions. The i-th question asks how many different travel plans will make shinobu visit the country x_i.

Input

The first line of the input contains one integer T $(1 \le T \le 5)$ — the number of test cases. Then T test cases follow.

For each testcase, the first line contains four integers $P, a, n, q(2 \le a < P \le 1000000007, 1 \le n \le 1000,$ $1 \le q \le 1000)$ — the number of countries, the value of a, the number of Shinobu's travel plans and the number of playf's questions.

Each of the next n lines contains two integers $s_i,\ d_i(0 \le s_i < P, 1 \le d_i \le 200000)$ — the starting country and the travel days.

Each of the next q lines contains one integer $x_i(0 \le x_i < P)$ — playf's questions.

It is guaranteed that P is a prime number.

Output

For each testcase, print q lines, the i-th line contains one integer — the answer to the i-th question.

Example

standard input	standard output

2
3 2 1 1
1 1
2
5 4 3 2
1 4
4 3
2 100000
4
2
	1
	2
	1

1008.Shinobu Loves Segment Tree

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

As a cold-blooded, hot-blooded, and iron-blooded vampire, Shinobu likes to build segment trees.

She uses the *build*() function to build a segment tree, and the process of building the segment tree will increase the value of some numbers.

The specific content of the *build*() function is as follows:

```
|$void\ build(int\ id,\ int\ l,\ int\ r):$
|    $value[id] += r - l + 1;$
|    $if(l == r)\ return;$
|    $int\ mid = (r + l)/2;$
|    $build(id * 2,\ l,\ mid);$
|    $build(id * 2 + 1,\ mid + 1,\ r);$
|    $return;$
```

For example, if Shinobi calls $build(1, 1, 2)$ once, then $value[1]$ will increase by 2, $value[2]$ and $value[3]$ will increase by 1.

In the long life of a vampire, Shinobu builds a segment tree every day. She has been doing this since day 1, and on the i_{th} day, she will call $build(1, 1, i)$ to build a segment tree.

As a fan of Shinobi, playf got a fan number x. Now he wants to ask you a question: what the $value[x]$ will be at the end of the n_{th} day ?

Input

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

Each of the next t lines contains two integers $n, x(1 \le n \le 10^9, 1 \le x \le 4 \times n)$ — playf's question.

Output

For each question, print one line contains one integer — the answer to the i_{th} question.

Examples

standard input	standard output

3
2 3
3 3
4 3
	1
	2
	4

1 26 49	9

3
1000000000 4000000000
1000000000 1
11451419 19810
	0
	500000000500000000
	4004478229

1009.Map

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	256 megabytes

Sakuyalove has a large world map M and a small world map m. Both of the them are in the shape of rectangle. The small map m is compressed from the large map M. If the length of M is a and the width of M is b, then the length of m is ka and the width of m is kb, where $0 < k < 1$. Now Sakuyalove puts the small map m on the large map M such that the small map is completely within the big map (including boundaries). She was surprised to find out that no matter how she places the small map, there always exists **exactly one** point P represents the same place in small map and large map (For example, in the following pictures, the location of the pin on both maps represents Tokyo, Japan). Sakuyalove wants to find out this point P. Please help her.

Input

The first line contains one integer $T(1 \le T \le 10^5)$, described the number of test cases.

Each test case contains eight lines. Each line has two integers $x, y(-10^3 \le x, y \le 10^3)$ separated by one space.

The first four lines are the coordinates of the upper left corner, the upper right corner, the lower right corner and the lower left corner of M.

The last four lines are the coordinates of the upper left corner, the upper right corner, the lower right corner and the lower left corner of m.

It is guaranteed that m is within M, both of the them are in the shape of rectangle, and m is compressed from M.

Please note that the upper left corner, the upper right corner, the lower right corner and the lower left corner of m and M are **one-to-one corresponding**. For example, in the picture of Note below, the correspondence of points is $A - a$, $B - b$, $C - c$, $D - d$. But $A - c$, $B - d$, $C - a$, $D - b$ is **not allowed**.

Output

Your output should contains T lines. Each line contains two real numbers x, y separated by one space, represents the coordinates of the point P. Your absolute error should not exceed 10^{-6}.

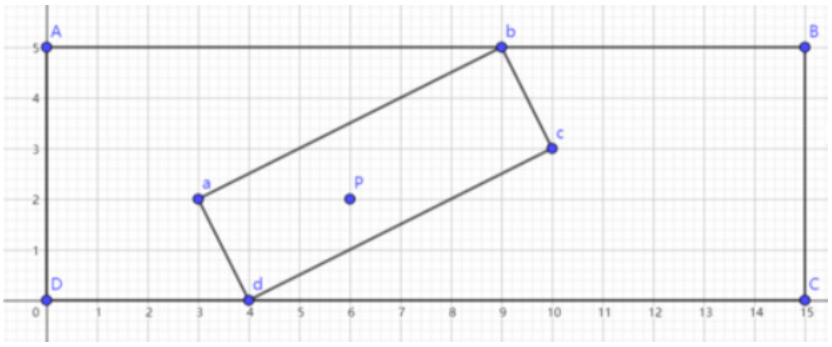
Example

standard input	standard output

1
0 5
15 5
15 0
0 0
3 2
9 5
10 3
4 0
	6.000000 2.000000

Note

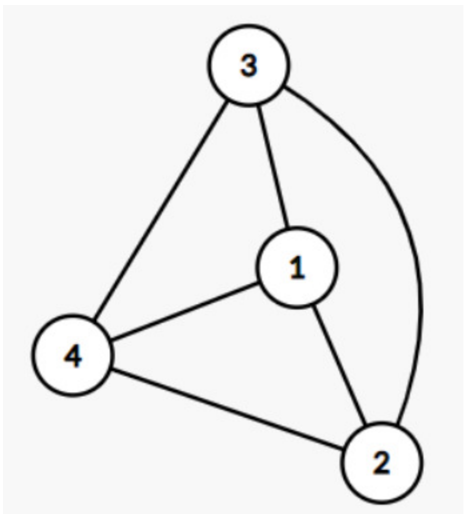
In the first example, the picture is like this:



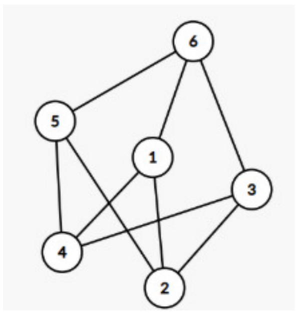
1010.Planar Graph

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

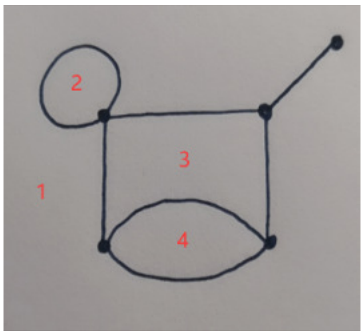
We say an undirected graph is a planar graph, if it exists a way to draw it on a planar, such that no two edges have intersection **except the endpoint**. For example, the graph below is a planar graph:



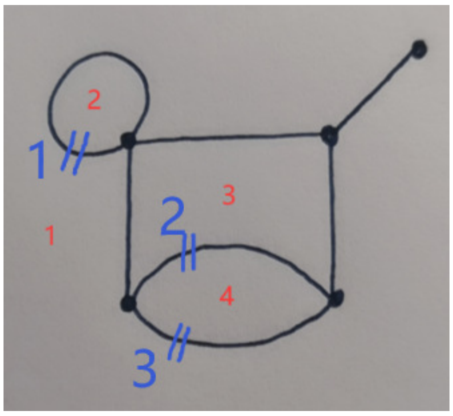
But this graph below is **not** a planar graph, since it can be proved that no matter how to draw this graph on a planar, at least two edges have intersection which is not an endpoint:



For a planar graph, it has some areas separated by edges. For example, the planar graph below has 4 areas (Note that the area 1 is the infinite planar outside):



Give you a planar graph with n vertices and m edges. Each area sets a country. You are the designer and you want to build some tunnels on the edges such that: From one city, you can travel to any other city by going through some tunnels or passing some cities(i.e. you can't cross one edge unless it sets a tunnel). For example, for the graph above, you can build tunnels like this:



In the picture above, you can travel from city 2 to city 3 by going through tunnel 1, passing the city 1, then going through tunnel 3, passing the city 4, finally going through tunnel 2, and you can arrive to city 3. You can check that from any city you can travel to any other city.

You want the number of tunnels as small as possible. Print the minimum number of tunnels and the ID of edges you build tunnel on.

Input

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

For each test case:

The first line contains two integers $n, m(1 \le n \le 10^5, 0 \le m \le 2 \times 10^5)$ separated by a space, described the number of vertices and edges.

Next m lines, the i-th line contains two integers $u, v(1 \le u, v \le n)$, separated by a space, described the endpoint of the i-th edge. The ID of the i-th edge is i.

It is guaranteed that the given graph is a planar graph. **But this graph may have self-loop, parallel edge and the graph may not connected**.

Output

The output contains $2T$ lines.

For each test case:

The first line contains one integer f, described the minimum number of tunnels you have to build.

The second lines contains f integers separated by spaces, the i-th integer described the ID of edges the i-th tunnel built on.

If for a fixed minimum number of tunnels, it has many ways to build the tunnels, **print the lexicographically smallest answer**.

Example

standard input	standard output

1
5 7
1 1
1 2
1 3
3 4
3 4
2 4
2 5
	3
	1 2 4

1011.Find different

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

You are given two integers n,m.

Two array $x = \{x_0, x_1, \ldots, x_{l-1}\}, y = \{y_0, y_1, \ldots,\ y_{l-1}\}$ of length l are considered **different** if x couldn't become y by performing the following operations any number of times:

- operation 1: Change x to b , for each i $(0 \le i < l), b_i \ = \ (x_i + 1)\ mod\ m$
- operation 2: Change x to b , for each i $(0 \le i < l), b_i \ = \ x_{(i+1)\ mod\ l}$

As an example, if $m = 3,\ l = 3$, $(0, 2, 2)$ and $(0, 1, 0)$ are considered **not different** because $(0, 2, 2)$ can become $(0, 1, 0)$ as follows: $(0, 2, 2) \overset{operation\ 1}{\longrightarrow} (1, 0, 0) \overset{operation\ 2}{\longrightarrow} (0, 0, 1) \overset{operation\ 2}{\longrightarrow} (0, 1, 0)$

For each i $(1 \le i \le n)$, find the number of different integer array a of length i, satisfied $\forall j \in (0, 1, \ldots, i-1), 0 \le a_j \le m-1$.

Since the answer may be too large, print it modulo 998244353.

Input

The first line of the input contains one integer T $(1 \le T \le 100)$ — the number of test cases. Then T testcases follow.

Each of the next T lines contains two integers n, m $(1 \le n, m \le 100000)$.

The sum of n over all testcases doesn't exceed 10^6.

Output

For each testcase,output one line contains n integers, separated by space, the i-th integer indicating the number of different a of length i, modulo 998244353.

Example

standard input	standard output

2
10 2
10 100000
	1 2 2 4 4 8 10 20 30 56
	1 50001 338600275 682529035 345997022 799071125 76757396

1012.Loop

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	512 megabytes

You are given an array a of length n. You must perform exactly k times operations.

For each operation,

- First, you select two integers l, r $(1 \le l \le r \le n)$,
- Second, change a to b, satisfy :
  - For each i $(1 \le i < l)$, $b_i = a_i$;
  - For each i $(l \le i < r)$, $b_i = a_{i+1}$;
  - $b_r = a_l$
  - For each i $(r < i \le n)$, $b_i = a_i$;

Find the lexicographically largest possible array after k times operations.

Array x is lexicographically greater than array y if there exists an index i $(1 \le i \le n)$ such that $x_i > y_i$ and for every j $(1 \le j < i)$, $x_j = y_j$.

Input

The first line of the input contains one integer T $(1 \le T \le 100$ $)$ — the number of test cases. Then T test cases follow.

The first line of the test case contains two integers n $(1 \le n, k \le 300000)$

The second line of the test case contains n integers $a_1, a_2, ..., a_n (1 \le a_i \le 300000)$

The sum of n over all testcases doesn't exceed 10^6.

The sum of k over all testcases doesn't exceed 10^6.

Output

For each testcase,one line contains n integers ,$a_1, a_2, ..., a_n$ — the lexicographically largest possible array after k times operations.

Example

standard input	standard output

2
7 3
1 4 2 1 4 2 4
5 2
4 3 5 4 5
	4 4 2 4 2 1 1
	5 4 5 4 3