A. Minimal Coprime

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

Today, Little John used all his savings to buy a segment. He wants to build a house on this segment.

A segment of positive integers $[l, r]$ is called *coprime* if l and r are coprime*.

A coprime segment $[l, r]$ is called *minimal coprime* if it does not contain† any coprime segment not equal to itself. To better understand this statement, you can refer to the notes.

Given $[l, r]$, a segment of positive integers, find the number of minimal coprime segments contained in $[l, r]$.

---

*Two integers a and b are coprime if they share only one positive common divisor. For example, the numbers 2 and 4 are not coprime because they are both divided by 2 and 1, but the numbers 7 and 9 are coprime because their only positive common divisor is 1.

†A segment $[l', r']$ is contained in the segment $[l, r]$ if and only if $l \le l' \le r' \le r$.

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 only line of each test case consists of two integers l and r ($1 \le l \le r \le 10^9$).

Output

For each test case, output the number of minimal coprime segments contained in $[l, r]$, on a separate line.

Standard Input	Standard Output
6 1 2 1 10 49 49 69 420 1 1 9982 44353	1 9 0 351 1 34371

Note

On the first test case, the given segment is $[1, 2]$. The segments contained in $[1, 2]$ are as follows.

- $[1, 1]$: This segment is coprime, since the numbers 1 and 1 are coprime, and this segment does not contain any other segment inside. Thus, $[1, 1]$ is minimal coprime.
- $[1, 2]$: This segment is coprime. However, as it contains $[1, 1]$, which is also coprime, $[1, 2]$ is not minimal coprime.
- $[2, 2]$: This segment is not coprime because 2 and 2 share 2 positive common divisors: 1 and 2.

Therefore, the segment $[1, 2]$ contains 1 minimal coprime segment.

,

B. Subsequence Update

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

After Little John borrowed expansion screws from auntie a few hundred times, eventually she decided to come and take back the unused ones.

But as they are a crucial part of home design, Little John decides to hide some in the most unreachable places — under the eco-friendly wood veneers.

You are given an integer sequence $a_1, a_2, \ldots, a_n$, and a segment $[l, r]$ ($1 \le l \le r \le n$).

You must perform the following operation on the sequence **exactly once**.

- Choose any **subsequence*** of the sequence a, and reverse it. Note that the subsequence does not have to be contiguous.

Formally, choose any number of indices $i_1, i_2, \ldots, i_k$ such that $1 \le i_1 < i_2 < \ldots < i_k \le n$. Then, change the i_x-th element to the original value of the i_{k-x+1}-th element simultaneously for all $1 \le x \le k$.

Find the **minimum value** of $a_l + a_{l+1} + \ldots + a_{r-1} + a_r$ after performing the operation.

---

*A sequence b is a subsequence of a sequence a if b can be obtained from a by the deletion of several (possibly, zero or all) element from arbitrary positions.

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 of each test case contains three integers n, l, r ($1 \le l \le r \le n \le 10^5$) — the length of a, and the segment $[l, r]$.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

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

Output

For each test case, output the minimum value of $a_l + a_{l+1} + \ldots + a_{r-1} + a_r$ on a separate line.

Standard Input	Standard Output
6 2 1 1 2 1 3 2 3 1 2 3 3 1 3 3 1 2 4 2 3 1 2 2 2 5 2 5 3 3 2 3 5	1 3 6 3 11 8

6 1 3 3 6 6 4 3 2	

Note

On the second test case, the array is $a = [1, 2, 3]$ and the segment is $[2, 3]$.

After choosing the subsequence a_1, a_3 and reversing it, the sequence becomes $[3, 2, 1]$. Then, the sum $a_2 + a_3$ becomes 3. It can be shown that the minimum possible value of the sum is 3.

,

C. Remove Exactly Two

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

Recently, Little John got a tree from his aunt to decorate his house. But as it seems, just one tree is not enough to decorate the entire house. Little John has an idea. Maybe he can remove a few vertices from the tree. That will turn it into more trees! Right?

You are given a tree* of n vertices. You must perform the following operation **exactly twice**.

- Select a vertex v;
- Remove all edges incident to v, and also the vertex v.

Please find the maximum number of connected components after performing the operation **exactly twice**.

Two vertices x and y are in the same connected component if and only if there exists a path from x to y. For clarity, note that the graph with 0 vertices has 0 connected components by definition.†

---

*A tree is a connected graph without cycles.

†But is such a graph connected?

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 of each test case contains a single integer n ($2 \le n \le 2 \cdot 10^5$).

Each of the next $n-1$ lines contains two integers u_i and v_i, denoting the two vertices connected by an edge ($1 \le u_i, v_i \le n$, $u_i \ne v_i$). It is guaranteed that the given edges form a tree.

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

Output

For each test case, output the maximum number of connected components on a separate line.

Standard Input	Standard Output
3 2 1 2 4 1 2 2 3 2 4 7 1 2 1 3 2 4 4 5 5 6 5 7	0 2 4

Note

On the first test case, removing a vertex twice will make the graph empty. By definition, the number of connected components in the graph with 0 vertices is 0. Therefore, the answer is 0.

On the second test case, removing two vertices 1 and 2 leaves 2 connected components. As it is impossible to make 3 connected components with 2 vertices, the answer is 2.

On the third test case, removing two vertices 1 and 5 leaves 4 connected components, which are $\{2, 4\}$, $\{3\}$, $\{6\}$, and $\{7\}$. It can be shown that it is impossible to make 5 connected components. Therefore, the answer is 4.

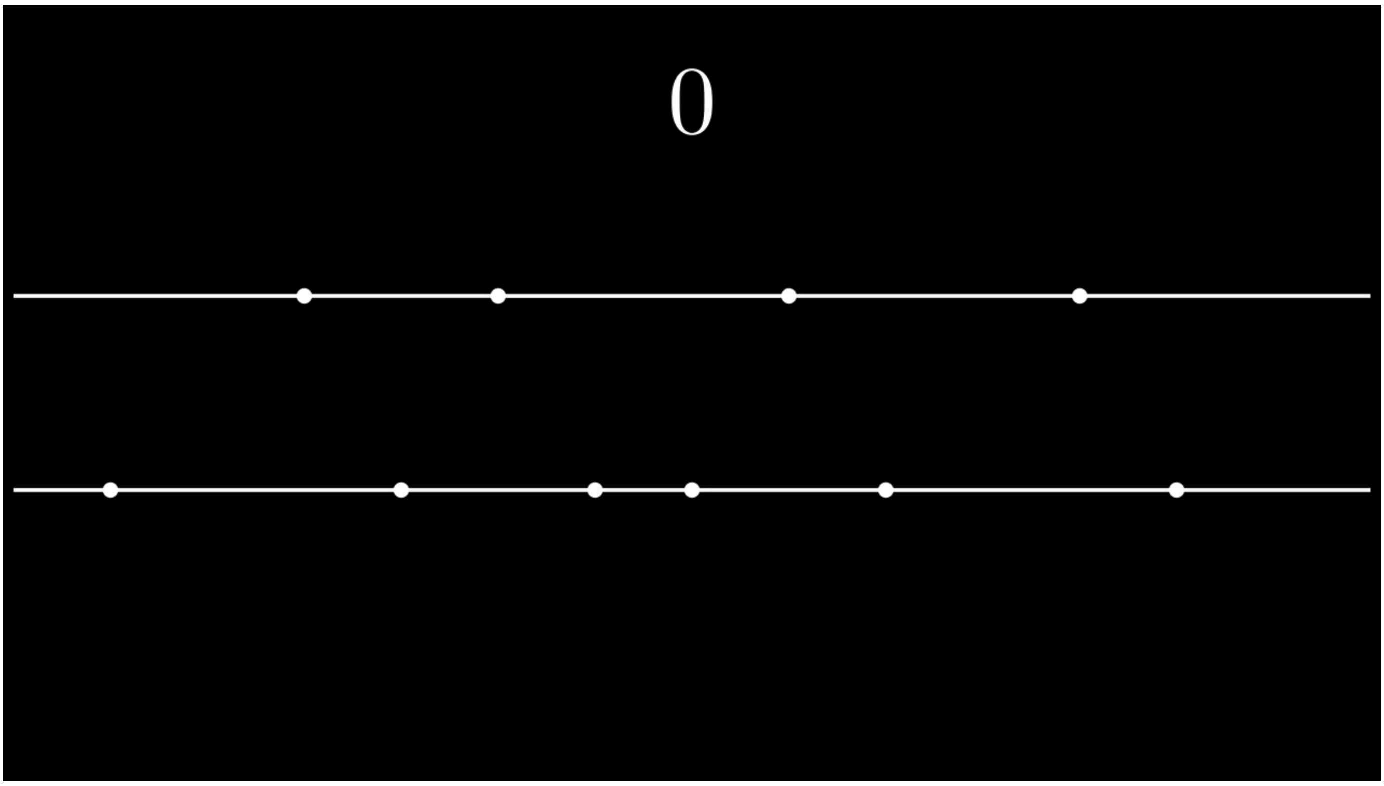
D. Game With Triangles

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

Even Little John needs money to buy a house. But he recently lost his job; how will he earn money now? Of course, by playing a game that gives him money as a reward! Oh well, maybe not those kinds of games you are thinking about.

There are $n+m$ distinct points $(a_1,0),(a_2,0),\ldots,(a_n,0),(b_1,2),(b_2,2),\ldots,(b_m,2)$ on the plane. Initially, your score is 0. To increase your score, you can perform the following operation:

- Choose three distinct points which are not collinear;
- Increase your score by the area of the triangle formed by these three points;
- Then, erase the three points from the plane.



An instance of the game, where the operation is performed twice.

Let $k_{\max}$ be the maximum number of operations that can be performed. For example, if it is impossible to perform any operation, $k_{\max}$ is 0. Additionally, define $f(k)$ as the maximum possible score achievable by performing the operation **exactly k times**. Here, $f(k)$ is defined for all integers k such that $0 \le k \le k_{\max}$.

Find the value of $k_{\max}$, and find the values of $f(x)$ for all integers $x=1,2,\ldots,k_{\max}$ independently.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 3 \cdot 10^4$). The description of the test cases follows.

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

The second line of each test case contains n pairwise distinct integers $a_1,a_2,\ldots,a_n$ — the points on $y=0$ ($-10^9 \le a_i \le 10^9$).

The third line of each test case contains m pairwise distinct integers $b_1, b_2, \ldots, b_m$ — the points on $y = 2$ ($-10^9 \le b_i \le 10^9$).

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

Output

For each test case, given that the maximum number of operations is $k_{\max}$, you must output at most two lines:

- The first line contains the value of $k_{\max}$;
- The second line contains $k_{\max}$ integers denoting $f(1), f(2), \ldots, f(k_{\max})$. You are allowed to omit this line if $k_{\max}$ is 0.

Note that under the constraints of this problem, it can be shown that all values of $f(x)$ are integers no greater than 10^{16}.

Standard Input	Standard Output
5 1 3 0 0 1 -1 2 4 0 100 -100 -50 0 50 2 4 0 1000 -100 -50 0 50 6 6 20 1 27 100 43 42 100 84 1 24 22 77 8 2 564040265 -509489796 469913620 198872582 -400714529 553177666 131159391 -20796763 -1000000000 1000000000	1 2 2 150 200 2 1000 200 4 99 198 260 283 2 2000000000 2027422256

Note

On the first test case, there are $1 + 3 = 4$ points $(0, 0), (0, 2), (1, 2), (-1, 2)$.

It can be shown that you cannot perform two or more operations. The value of $k_{\max}$ is 1, and you are only asked for the value of $f(1)$.

You can choose $(0, 0)$, $(-1, 2)$, and $(1, 2)$ as the three vertices of the triangle. After that, your score is increased by the area of the triangle, which is 2. Then, the three points are erased from the plane. It can be shown that the maximum value of your score after performing one operation is 2. Therefore, the value of $f(1)$ is 2.

On the fifth test case, there are $8 + 2 = 10$ points.

It can be shown that you cannot perform three or more operations. The value of $k_{\max}$ is 2, and you are asked for the values $f(1)$ and $f(2)$.

To maximize the score with only one operation, you can choose three points $(198\,872\,582, 0)$, $(-1\,000\,000\,000, 2)$, and $(1\,000\,000\,000, 2)$. Then, the three points are erased from the plane. It can be shown that the maximum value of your score after performing one operation is $2\,000\,000\,000$. Therefore, the value of $f(1)$ is $2\,000\,000\,000$.

To maximize the score with exactly two operations, you can choose the following sequence of operations.

- Choose three points $(-509\,489\,796, 0)$, $(553\,177\,666, 0)$, and $(-1\,000\,000\,000, 2)$. The three points are erased.
- Choose three points $(-400\,714\,529, 0)$, $(564\,040\,265, 0)$, and $(1\,000\,000\,000, 2)$. The three points are erased.

Then, the score after two operations becomes $2\,027\,422\,256$. It can be shown that the maximum value of your score after performing exactly two operations is $2\,027\,422\,256$. Therefore, the value of $f(2)$ is $2\,027\,422\,256$.

,

E. Triangle Tree

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

One day, a giant tree grew in the countryside. Little John, with his childhood eagle, decided to make it his home. Little John will build a structure on the tree with galvanized square steel. However, little did he know, he could not build what is physically impossible.

You are given a rooted tree[*] containing n vertices rooted at vertex 1. A pair of vertices (u, v) is called a *good pair* if u is not an ancestor[†] of v and v is not an ancestor of u. For any two vertices, $\text{dist}(u, v)$ is defined as the number of edges on the unique simple path from u to v, and $\text{lca}(u, v)$ is defined as their lowest common ancestor.

A function $f(u, v)$ is defined as follows.

- If (u, v) is a good pair, $f(u, v)$ is the number of distinct integer values x such that there exists a **non-degenerate triangle**[‡] formed by side lengths $\text{dist}(u, \text{lca}(u, v))$, $\text{dist}(v, \text{lca}(u, v))$, and x.
- Otherwise, $f(u, v)$ is 0.

You need to find the following value:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f(i, j).$$

---

[*] A tree is a connected graph without cycles. A rooted tree is a tree where one vertex is special and called the root.

[†] An ancestor of vertex v is any vertex on the simple path from v to the root, including the root, but not including v. The root has no ancestors.

[‡] A triangle with side lengths a, b, c is non-degenerate when $a + b > c$, $a + c > b$, $b + c > a$.

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 of each test case contains a single integer n ($1 \le n \le 3 \cdot 10^5$).

Each of the next $n - 1$ lines contains two integers u_i and v_i, denoting the two vertices connected by an edge ($1 \le u_i, v_i \le n$, $u_i \ne v_i$).

It is guaranteed that the given edges form a tree.

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, output the answer on a separate line.

Standard Input	Standard Output
4 3 1 2	1 0

1 3 3 1 2 3 2 5 2 3 1 5 4 2 1 2 11 2 1 2 3 2 4 4 5 6 5 5 7 4 8 8 9 7 10 10 11	4 29

Note

On the first test case, the only good pair (i, j) satisfying $i < j$ is $(2, 3)$. Here, $\mathrm{lca}(2, 3)$ is 1, and the two distances are 1 and 1.

There is only one value of x for two side lengths 1 and 1, which is 1. Therefore, the answer for the first test case is 1.

On the second test case, there is no good pair. Therefore, the answer for the second test case is 0.

On the third test case, the good pairs (i, j) satisfying $i < j$ are as follows.

- $(2, 5)$: $\mathrm{lca}(2, 5)$ is 1, distances are 1 and 1. There is only one possible value of x, which is 1.
- $(3, 4)$: $\mathrm{lca}(3, 4)$ is 2, distances are 1 and 1. There is only one possible value of x, which is 1.
- $(3, 5)$: $\mathrm{lca}(3, 5)$ is 1, distances are 2 and 1. There is only one possible value of x, which is 2.
- $(4, 5)$: $\mathrm{lca}(4, 5)$ is 1, distances are 2 and 1. There is only one possible value of x, which is 2.

Therefore, the answer for the third test case is $1 + 1 + 1 + 1 = 4$.

,

F1. Counting Is Not Fun (Easy Version)

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

This is the easy version of the problem. The difference between the versions is that in this version, the limits on t and n are smaller. You can hack only if you solved all versions of this problem.

Now Little John is rich, and so he finally buys a house big enough to fit himself and his favorite bracket sequence. But somehow, he ended up with a lot of brackets! Frustrated, he penetrates through the ceiling with the "buddha palm".

A bracket sequence is called *balanced* if it can be constructed by the following formal grammar.

1. The empty sequence $\varnothing$ is balanced.
2. If the bracket sequence A is balanced, then (A) is also balanced.
3. If the bracket sequences A and B are balanced, then the concatenated sequence AB is also balanced.

For example, the sequences "$(())()$", "$()$", "$(()(()))$", and the empty sequence are balanced, while "$(()$" and "$(()))($" are not.

Given a balanced bracket sequence s, a pair of indices (i, j) $(i < j)$ is called a *good pair* if s_i is '$($', s_j is '$)$', and the two brackets are added simultaneously with respect to Rule 2 while constructing the sequence s. For example, the sequence "$(())()$" has three different good pairs, which are $(1, 4)$, $(2, 3)$, and $(5, 6)$. One can show that any balanced bracket sequence of $2n$ brackets contains exactly n different good pairs, and using any order of rules to construct the same bracket sequence will yield the same set of good pairs.

Emily will play a bracket guessing game with John. The game is played as follows.

Initially, John has a balanced bracket sequence s containing n different good pairs, which is not known to Emily. John tells Emily the value of n and asks Emily to guess the sequence.

Throughout n turns, John gives Emily the following kind of clue on each turn.

- l r: The sequence s contains a good pair (l, r).

The clues that John gives Emily are pairwise distinct and do not contradict each other.

At a certain point, Emily can be certain that the balanced bracket sequence satisfying the clues given so far is unique. For example, assume Emily knows that s has 3 good pairs, and it contains the good pair $(2, 5)$. Out of 5 balanced bracket sequences with 3 good pairs, there exists only one such sequence "$((()))$" with the good pair $(2, 5)$. Therefore, one can see that Emily does not always need n turns to guess s.

To find out the content of s as early as possible, Emily wants to know the number of different balanced bracket sequences that match the clues after each turn. Surely, this is not an easy job for Emily, especially when she is given so many good pairs. Now it is your turn to help Emily. Given the clues, you must find the answer before and after each turn. As the answers may be huge, you need to find them modulo $998\,244\,353$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^3)$. The description of the test cases follows.

The first line of each test case contains one integer n $(2 \le n \le 5000)$ — the number of good pairs.

Then, each of the n following lines contains two integers l_i and r_i representing the i-th clue ($1 \le l_i < r_i \le 2n$).

The clues in one test case are pairwise distinct and do not contradict each other.

It is guaranteed that the sum of n over all test cases does not exceed 5000.

Output

For each test case, output $n+1$ integers on a separate line:

- The first integer is the answer before all clues, modulo $998\,244\,353$.
- For all $i \ge 1$, the $i+1$-th integer is the answer after the i-th clue, modulo $998\,244\,353$.

Standard Input	Standard Output
3 3 2 5 1 6 3 4 4 1 6 7 8 2 3 4 5 6 2 3 1 6 7 8 9 12 10 11 4 5	5 1 1 1 14 2 2 1 1 132 42 5 2 1 1 1

Note

The first test case of the example is explained in the problem description.

The third test case of the example is explained as follows. It can be shown that there are 132 balanced bracket sequences with 6 good pairs. The answers after each clue are given as follows:

1. You are given the good pair $(2,3)$. There are 42 balanced bracket sequences having the good pair $(2,3)$.
2. You are given the good pair $(1,6)$. There are 5 balanced bracket sequences having good pairs $(2,3)$, $(1,6)$.
3. You are given the good pair $(7,8)$. There are 2 balanced bracket sequences having the three good pairs. The strings are "$(()())()(())$" and "$(()())()()()$", respectively.
4. You are given the good pair $(9,12)$. There is only one balanced bracket sequence having the four good pairs. The content of s is therefore the only string, which is "$(()())()(())$".

Then, the number of bracket sequences after the fifth and the sixth clue are both 1 as you already know the content of s.

,

F2. Counting Is Not Fun (Hard Version)

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

This is the hard version of the problem. The difference between the versions is that in this version, the limits on t and n are bigger. You can hack only if you solved all versions of this problem.

Now Little John is rich, and so he finally buys a house big enough to fit himself and his favorite bracket sequence. But somehow, he ended up with a lot of brackets! Frustrated, he penetrates through the ceiling with the "buddha palm".

A bracket sequence is called *balanced* if it can be constructed by the following formal grammar.

1. The empty sequence $\varnothing$ is balanced.
2. If the bracket sequence A is balanced, then (A) is also balanced.
3. If the bracket sequences A and B are balanced, then the concatenated sequence AB is also balanced.

For example, the sequences "(())()", "()", "(()(()))", and the empty sequence are balanced, while "(()" and "(()))(" are not.

Given a balanced bracket sequence s, a pair of indices (i, j) ($i < j$) is called a *good pair* if s_i is '(', s_j is ')', and the two brackets are added simultaneously with respect to Rule 2 while constructing the sequence s. For example, the sequence "(())()" has three different good pairs, which are $(1, 4)$, $(2, 3)$, and $(5, 6)$. One can show that any balanced bracket sequence of $2n$ brackets contains exactly n different good pairs, and using any order of rules to construct the same bracket sequence will yield the same set of good pairs.

Emily will play a bracket guessing game with John. The game is played as follows.

Initially, John has a balanced bracket sequence s containing n different good pairs, which is not known to Emily. John tells Emily the value of n and asks Emily to guess the sequence.

Throughout n turns, John gives Emily the following kind of clue on each turn.

- l r: The sequence s contains a good pair (l, r).

The clues that John gives Emily are pairwise distinct and do not contradict each other.

At a certain point, Emily can be certain that the balanced bracket sequence satisfying the clues given so far is unique. For example, assume Emily knows that s has 3 good pairs, and it contains the good pair $(2, 5)$. Out of 5 balanced bracket sequences with 3 good pairs, there exists only one such sequence "((()))" with the good pair $(2, 5)$. Therefore, one can see that Emily does not always need n turns to guess s.

To find out the content of s as early as possible, Emily wants to know the number of different balanced bracket sequences that match the clues after each turn. Surely, this is not an easy job for Emily, especially when she is given so many good pairs. Now it is your turn to help Emily. Given the clues, you must find the answer before and after each turn. As the answers may be huge, you need to find them modulo $998\,244\,353$.

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 of each test case contains one integer n ($2 \le n \le 3 \cdot 10^5$) — the number of good pairs.

Then, each of the n following lines contains two integers l_i and r_i representing the i-th clue ($1 \leq l_i < r_i \leq 2n$).

The clues in one test case are pairwise distinct and do not contradict each other.

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, output $n + 1$ integers on a separate line:

- The first integer is the answer before all clues, modulo $998\,244\,353$.
- For all $i \geq 1$, the $i + 1$-th integer is the answer after the i-th clue, modulo $998\,244\,353$.

Standard Input	Standard Output
3 3 2 5 1 6 3 4 4 1 6 7 8 2 3 4 5 6 2 3 1 6 7 8 9 12 10 11 4 5	5 1 1 1 14 2 2 1 1 132 42 5 2 1 1 1

Note

The first test case of the example is explained in the problem description.

The third test case of the example is explained as follows. It can be shown that there are 132 balanced bracket sequences with 6 good pairs. The answers after each clue are given as follows:

1. You are given the good pair $(2, 3)$. There are 42 balanced bracket sequences having the good pair $(2, 3)$.
2. You are given the good pair $(1, 6)$. There are 5 balanced bracket sequences having good pairs $(2, 3)$, $(1, 6)$.
3. You are given the good pair $(7, 8)$. There are 2 balanced bracket sequences having the three good pairs. The strings are "$(()())()(())$" and "$(()())()()()$", respectively.
4. You are given the good pair $(9, 12)$. There is only one balanced bracket sequence having the four good pairs. The content of s is therefore the only string, which is "$(()())()(())$".

Then, the number of bracket sequences after the fifth and the sixth clue are both 1 as you already know the content of s.