Problem A. Max-Min Madness

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

You are given an array a of length n. You can perform the following two operations any number of times, in any order:

1. Choose an index i such that $a_i > 0$, and decrement a_i by 1. In other words, assign $a_i := a_i - 1$.
2. Select $k \geq 2$ distinct indices $i_1, i_2, \ldots, i_k$ such that there exists at least one pair of indices i_x and i_y satisfying $a_{i_x} \neq a_{i_y}$. In other words, not all the values in the picked indices are the same. Then simultaneously set the values of all of $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ to $(\max(a_{i_1}, a_{i_2}, \ldots, a_{i_k}) - \min(a_{i_1}, a_{i_2}, \ldots, a_{i_k}) - 1)$.

   For example, given $a = [1, 0, 1, 9, 1]$, choosing a_1, a_3, a_4 for operation 2 results in $a = [7, 0, 7, 7, 1]$. However, choosing a_1, a_3, a_5 for operation 2 is invalid, as their values are the same.

You want to choose a sequence of operations that maximizes the total number of operations. Find the maximum number of operations possible. Since the answer can be large, output it modulo 998 244 353.

Input

The first line contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases.

For each test case, the first line contains an integer n ($1 \leq n \leq 2 \cdot 10^5$) — the size of the array a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 10^6$).

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 a single integer in a new line — the maximum number of operations possible modulo 998 244 353.

Example

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

Note

In the first test case, you can only choose a_1 for operation 1 five times.

In the second test case, the optimal strategy is the following:

1. Choose a_1 and a_2 for operation 2. After that, $a = [1, 1]$.
2. Choose a_1 for operation 1. After that, $a = [0, 1]$.
3. Choose a_2 for operation 1. After that, $a = [0, 0]$.

After this, we can no longer perform any of the two operations on the array.

Problem B. The Absolute MEX Challenge

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

Given n, please construct a permutation* p of length n such that $\text{MEX}^{\dagger}(\{|p_1 - 1|, |p_2 - 2|, \ldots, |p_n - n|\}) \geq n - 1$.

*A permutation is a sequence of integers from 1 to n of length n containing each number exactly once. For example, $[2, 1, 3]$ is a permutation, but $[2, 1, 2]$ and $[2, 1, 4]$ are not.

†The minimum excluded (MEX) of a set of integers is the smallest non-negative integer that does not appear in the set. For example, $\text{MEX}(\{1, 2, 3\}) = 0$ and $\text{MEX}(\{2, 9, 0, 1\}) = 3$.

Input

The first line contains a single integer t ($1 \leq t \leq 1000$) — the number of test cases.

Each test case consists of a single integer n ($1 \leq n \leq 3 \cdot 10^5$).

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

Output

For each test case, print n integers — a permutation that satisfies the aforementioned condition or -1 if no such permutation exists. If there are multiple solutions, output any.

Example

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

Note

In the first test case, for $n = 8$, a possible permutation is $[6, 8, 5, 4, 2, 7, 3, 1]$. The absolute differences are $\{|6 - 1|, |8 - 2|, |5 - 3|, |4 - 4|, |2 - 5|, |7 - 6|, |3 - 7|, |1 - 8|\} = \{5, 6, 2, 0, 3, 1, 4, 7\}$. The MEX of this set is $8 \geq n - 1 = 7$.

In the second test case, for $n = 5$, a possible permutation is $[5, 2, 4, 1, 3]$. The absolute differences are $\{|5 - 1|, |2 - 2|, |4 - 3|, |1 - 4|, |3 - 5|\} = \{4, 0, 1, 3, 2\}$. The MEX of this set is $5 \geq n - 1 = 4$.

In the third test case, for $n = 6$, a possible permutation is $[4, 6, 1, 3, 5, 2]$. The absolute differences are $\{|4 - 1|, |6 - 2|, |1 - 3|, |3 - 4|, |5 - 5|, |2 - 6|\} = \{3, 4, 2, 1, 0, 4\}$. The MEX of this set is $5 \geq n - 1 = 5$.

Problem C. Palindromic Palindrome Partition

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

You are given a string s of length n.

A partition of a string s is an ordered sequence of k contiguous non-empty substrings* $t_1, t_2, \ldots, t_k$ such that $t_1 + t_2 + \cdots + t_k = s$, where $+$ represents string concatenation.

A partition is considered valid if it satisfies both of the following conditions:

1. Each substring t_i in the partition is a palindrome†.
2. The sequence of substring lengths $|t_1|, |t_2|, \ldots, |t_k|$ also forms a palindrome‡.

The score of a valid partition is defined as k^2, where k is the number of substrings in the partition.

For example, if the string is `abaccded`, then a valid partition is `[aba, cc, ded]` because, `aba`, `cc`, and `ded` are all palindromes, and the sequence of lengths $[3, 2, 3]$ is also a palindrome. The score of this partition is $3^2 = 9$.

Your task is to compute the sum of scores over all valid partitions of s. Since the answer can be large, output it modulo 998 244 353.

*A string a is a substring of a string b if a can be obtained from b by deletion of several (possibly, zero or all) characters from the beginning and several (possibly, zero or all) characters from the end. For example, `stiu` is a substring of `austiupc`, but `stup` and `cpu` are not.

†A string is a palindrome if it reads the same forwards and backwards. For example, `abba` and `yobananaboy` are palindromes, but `abca` is not.

‡A sequence a of length n is a palindrome if $a_i = a_{n-i+1}$ for all i from 1 to n. For example, $[1, 2, 10, 2, 1]$ is a palindrome, but $[1, 2, 3, 4, 1]$ is not.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases.

The first and only line of each test cases contains a single string s ($1 \le |s| \le 5000$).

It is guaranteed that the sum of string lengths $|s|$ over all test cases does not exceed 5000.

Output

For each test case, print a single integer — the sum of scores over all valid partitions modulo 998 244 353.

Example

standard input	standard output
5 a aa ab aaxzz babccbab	1 5 4 34 173

Note

In the fourth test case, the valid partitions are:

- `[aa, x, zz]`, score: $3^2 = 9$. Here, `aa`, `x`, and `zz` are all palindromes, and the sequence of lengths $[2, 1, 2]$ is also a palindrome.
- `[a, a, x, z, z]`, score: $5^2 = 25$. Here, `a`, `a`, `x`, `z`, and `z` are all palindromes, and the sequence of lengths $[1, 1, 1, 1, 1]$ is also a palindrome.

The sum of all scores is $9 + 25 = 34$.

In the fifth test case, the valid partitions are:

- `[babccbab]`, score: $1^2 = 1$,
- `[b, abccba, b]`, score: $3^2 = 9$,
- `[b, a, bccb, a, b]`, score: $5^2 = 25$,
- `[b, a, b, cc, b, a, b]`, score: $7^2 = 49$,
- `[b, a, b, c, c, b, a, b]`, score: $8^2 = 64$,
- `[bab, cc, bab]`, score: $3^2 = 9$,
- `[bab, c, c, bab]`, score: $4^2 = 16$.

The sum of all scores is $1 + 9 + 25 + 49 + 64 + 9 + 16 = 173$.

Problem D. Strong Tree

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

If you seek a tree, there was no bluff!
Up for it, or are you not strong enough?

There is a rooted tree, consisting of n vertices numbered from 1 to n, and the root is the vertex 1. Each vertex i has a value a_i and a unique rank r_i.

The *strength* of a simple path from vertex u to vertex v is the sum of values of each of the vertices on the unique simple path from u to v.

For each vertex i from 1 to n, find the maximum *strength* of any simple path from u to v that satisfies the following conditions:

- Vertex i is an ancestor of both u and v.
- The simple path from u to v goes through vertex i.
- $r_j \leq r_i$ for any vertex j on the simple path from u to v.

Input

The first line contains an integer t ($1 \leq t \leq 10^4$) — number of test cases.

The first line of each test case contains an integer n ($2 \leq n \leq 10^5$) — the number of vertices.

The second line contains $n-1$ integers $p_2, p_3, \ldots, p_n$ ($1 \leq p_i \leq n$; $p_i \neq i$), where p_i is the parent of vertex i.

The third line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^7 \leq a_i \leq 10^7$), where a_i is the value of vertex i.

The fourth line contains n integers $r_1, r_2, \ldots, r_n$ ($1 \leq r_i \leq n$), where r_i is the rank of vertex i. It is guaranteed that $r_i \neq r_j$ for all $i \neq j$.

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

Output

For each test case print n integers on a single line — the i-th integer represents the answer for vertex i.

Example

standard input	standard output
2 4 1 1 2 6 5 9 7 3 1 2 4 6 1 1 3 3 3 5 2 -9 4 -16 36 6 1 4 2 3 5	20 5 9 7 34 2 -5 4 -16 36

Note

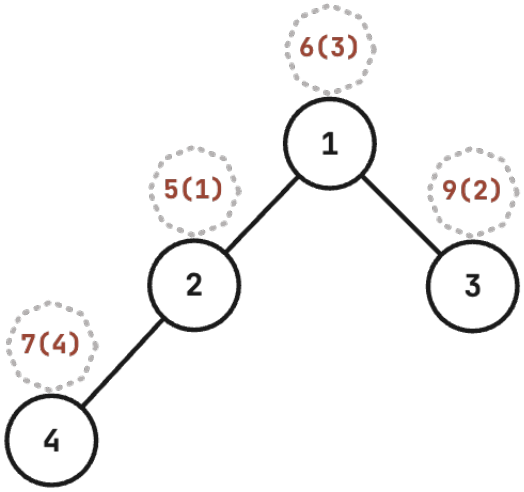


Figure: Tree from the 1st test case in sample

Each vertex i is labeled above it as $a_i(r_i)$ in the figure.

In the 1st test case, for vertex 1, the optimal path is from vertex 2 to vertex 3, since the path satisfies all the required conditions. We cannot include vertex 4 in the path since it has a higher rank than vertex 1.

For the remaining vertices, the optimal path includes only the vertex itself.

Problem E. Aloy and the Forbidden Code

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

In the world of *Horizon Forbidden West*, ancient secrets lie hidden beneath layers of code, and Aloy, a skilled machine hunter and seeker of truth, has just discovered a cryptic message. The message is encoded in a string, s, composed only of three distinct symbols: 'a', 'b', and 'c'. Aloy knows two crucial details about this encoded message:

1. No two adjacent characters are the same, i.e., $s_i \neq s_{i+1}$ for $1 \leq i \leq |s| - 1$ where $|s|$ is the length of the string.
2. Each of the three characters 'a', 'b', and 'c' appears at least once in the string.

To decipher this code and uncover its meaning, Aloy must extract the shortest contiguous substring* of s that contains all three characters at least once. Can you help Aloy break the code and determine the minimum length of such a substring?

*A string a is a substring of a string b if a can be obtained from b by deletion of several (possibly, zero or all) characters from the beginning and several (possibly, zero or all) characters from the end. For example, `riz` is a substring of `horizon`, but `rizz` and `ozon` are not.

Input

The first and only line contains a string s ($3 \leq |s| \leq 10^5$), consisting only of the characters 'a', 'b', and 'c'.

It is guaranteed that no two adjacent characters in s are the same, and that all three characters appear at least once in s.

Output

Print a single integer — the length of the shortest contiguous substring of s that contains all three characters.

Example

standard input	standard output
abac	3

Note

For the string `abac`, the shortest substring that contains all three characters is `bac`, which has a length of 3.

Problem F. Rotating Painter

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

Tamim is renovating his house, and there are many chores to be done. One of them is painting the old furniture. Painting furniture seemed simple — until he came across a peculiar table.

The tabletop consists of two boards: a top board and a bottom board. Both boards are regular convex polygons*. The top board is stacked on top of the bottom board. The boards are fixed at the same center† by a bolt.

The top board can rotate freely around its center, but it cannot be removed or repositioned. The bottom board remains stationary.

Tamim can easily paint the top board, but the bottom board is partially or completely covered by the top one. So it's hard to paint the bottom board. However, by rotating the top board, different portions of the bottom board become visible.

Tamim has an idea! He decides to paint the visible areas of the bottom board. To do this, he will do the following:

1. Paints the currently visible areas of the bottom board.
2. Rotates the top board by an arbitrary angle.
3. Paints any newly visible areas of the bottom board.
4. Repeats this process until no additional areas can be painted.

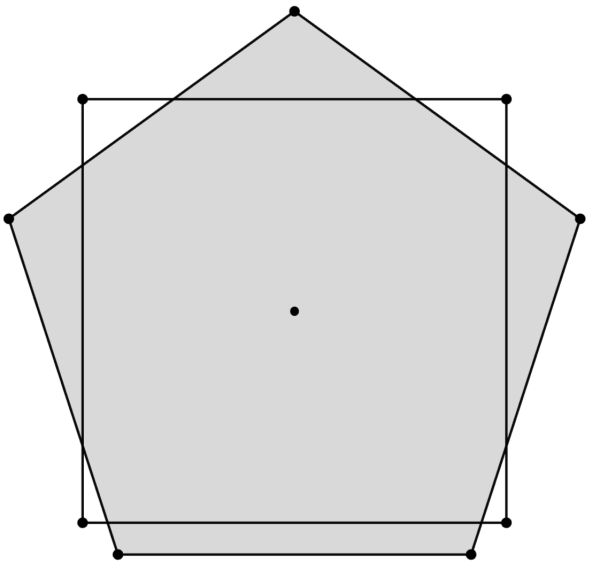


Figure: The pentagon is the top board and the square is the bottom board. Both have the same center.

After completing this process, Tamim calculates the total area of the bottom board that was successfully painted. Please help Tamim determine the maximum total area of the bottom board that can be painted.

*A regular convex polygon is a polygon where all the side lengths are equal, all the interior angles are equal, and all the interior angles are strictly less than 180°.

†The center of a regular convex polygon is the point equidistant from all its vertices.

Note: An animation has been provided to help you visualize the problem. Please check the attachments to view the animation.

Input

The first line contains an integer t ($1 \leq t \leq 10^5$) — the number of test cases.

The first line of each test case contains two integers n_t and a_t ($3 \leq n_t \leq 1000$; $1 \leq a_t \leq 1000$) where n_t is the number of sides of the top polygon and a_t is the length of each side of the top polygon.

The second line of each test case contains two integers n_b and a_b ($3 \leq n_b \leq 1000$; $1 \leq a_b \leq 1000$) where n_b is the number of sides of the bottom polygon and a_b is the length of each side of the bottom polygon.

Output

For each test case, output a single number — the maximum total area of the bottom board that can be painted.

Your answer will be considered correct if its absolute or relative error doesn't exceed 10^{-4}.

Example

standard input	standard output
6 5 5 4 6 10 6 11 5 11 6 13 8 10 10 12 8 3 1 3 1 1000 1000 1000 1000	2.820131 0.000000 515.943499 0.201572 0.171213 261798.698770

Problem G. GCD and LCM in Perfect Sync

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

Given a_1 and n, find the number of sequences of positive integers $a_2, a_3, \ldots, a_n$ that satisfy the following condition:

$$\frac{\text{lcm}(a_1, a_2, \ldots, a_n)}{\gcd(a_1, a_2, \ldots, a_n)} = a_1$$

Here, $\gcd(a_1, a_2, \ldots, a_n)$ denotes the greatest common divisor, which is the largest positive integer that divides each number in the sequence, and $\text{lcm}(a_1, a_2, \ldots, a_n)$ denotes the least common multiple, which is the smallest positive integer that is divisible by each number in the sequence.

Since the answer can be large, output it modulo 998 244 353. It can be proven that the answer is finite.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases.

The only line of each test case consists of two integers a_1 and n ($2 \le a_1, n \le 10^9$).

Output

For each test case, output the number of sequences satisfying the condition above modulo 998 244 353.

Example

standard input	standard output
4 2 3 4 3 666 999 1000000000 1000000000	6 12 15861573 208999360

Note

In the first test case, all the 6 valid sequences are $[2, 1, 1]$, $[2, 1, 2]$, $[2, 2, 1]$, $[2, 2, 4]$, $[2, 4, 2]$, and $[2, 4, 4]$.

Problem H. Flip to Zero

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

You are given two integers n and k. For an integer m $(1 \le m \le n)$, consider a binary string of length n, where the first $n - m$ characters are 0 and the last m characters are 1. You can perform the following operation on it any number of times:

- Select exactly k distinct indices of the string. Then flip all bits at the selected indices (i.e., change 0 to 1, and 1 to 0).

Let $f(m)$ be the minimum number of operations required to make all bits in the string 0, or -1 if it is impossible.

Your task is to compute $f(m)$ for all m from 1 to n.

Input

The first line contains a single integer t $(1 \le t \le 10^4)$ — the number of test cases.

The first and only line of each test case contains two integers n and k $(1 \le k \le n \le 3 \cdot 10^5)$.

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 integers — the m-th integer represents $f(m)$.

Example

standard input	standard output
3 4 1 3 3 7 5	1 2 3 4 -1 -1 1 3 2 3 2 1 4 3

Note

In the third test case, we have $n = 7$ and $k = 5$. For $m = 3$, the string looks like 0000111. For this string, we can perform the following operations:

1. Choose indices $\{1, 3, 4, 6, 7\}$ in the first operation. The string becomes 1011100.
2. Choose indices $\{2, 3, 5, 6, 7\}$ in the second operation. The string becomes 1101011.
3. Choose indices $\{1, 2, 4, 6, 7\}$ in the third operation. The string becomes 0000000.

So we can make all bits 0 in 3 operations. It can be shown that it is impossible to achieve this in less than 3 operations. So $f(3) = 3$.

In the second test case we have $n = 3$, and $k = 3$. For $m = 1$, the string looks like 001. For this string, in one operation you can only flip all the bits together. So after one operation, the string becomes 110, after two operations, the string becomes 001 again. So it is impossible to make all bits 0 in this case. So $f(1) = -1$.

Problem I. The Art of Rearrangement

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

Given an array a of length n, please rearrange the elements of a such that the values $a_i - i$ are pairwise distinct over all $1 \le i \le n$.

Input

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

For each test case, the first line contains a single integer n ($2 \le n \le 10^5$).

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

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

Output

For each test case, print n integers — a valid rearrangement of the array, or -1 if no such rearrangement exists. If there are multiple solutions, output any.

Example

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

Note

In the first test case, one valid rearrangement of the array is $[1, 3, 5, 7]$. Because, the differences are: $[a_1 - 1, a_2 - 2, a_3 - 3, a_4 - 4] = [0, 1, 2, 3]$. All of them are distinct.

Problem J. No Duplicates

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

You are given two integers n and m, and q restriction of the form (l_i, r_i).

For each k from 1 to q, count, modulo 998 244 353, the number of sequences of positive integers $a_1, a_2, \ldots, a_n$ such that the following conditions are satisfied:

- $1 \le a_i \le m$ for each i from 1 to n.
- For each i from 1 to k, all elements in the subarray $a_{l_i}, a_{l_i+1}, \ldots, a_{r_i}$ are distinct. That is, for each pair of indices $l_i \le x < y \le r_i$, $a_x \ne a_y$ is satisfied.

Input

The first line contains a single integer t ($1 \le t \le 5 \cdot 10^4$) — the number of test cases.

The first line of each test case contains three integers n, m and q ($1 \le n, q \le 3 \cdot 10^5, n \le m \le 10^6$).

The i-th of the next q lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$).

It is guaranteed that the sum of q over all test cases does not exceed $3 \cdot 10^5$. Note that there are no constraints on the sum of n or m.

Output

For each test case, print q integers on separate lines — the k-th integer represents the answer for k modulo 998 244 353.

Example

standard input	standard output
2 3 10 2 1 2 2 3 4 6 3 1 2 3 4 1 4	900 810 1080 900 360

Problem K. Primal Brackets

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

A sequence of integers is balanced if it satisfies the following rules:

- An empty sequence is balanced.
- If A and B are balanced sequences, then their concatenation AB is balanced.
- If A is a balanced sequence, then for any positive integer x, the sequence obtained by simultaneously adding $+x$ at the beginning of A and $-x$ at the end of A is balanced.

For example, the sequence $[1, 2, -2, -1]$ is balanced, but $[-1, 1, 2, -2]$ and $[1, 1]$ are not.

A balanced sequence of length $2k$ is called primal if the first k integers are positive and the last k integers are negative. For example, the sequence $[3, 2, 1, -1, -2, -3]$ is primal, but $[1, -1, 2, -2]$ is not.

For a balanced sequence b, let $f(b)$ be the minimum number of non-empty subsequences* into which b can be partitioned† into such that each subsequence is a primal sequence.

For example, for the balanced sequence $b = [1, 2, -2, 3, -3, -1]$, one possible partition into a minimal number of primal subsequences is: $[1, 2, -2, -1]$ and $[3, -3]$. Therefore, $f(b) = 2$.

You are given a balanced sequence $a_1, a_2, \ldots, a_{2n}$ where all elements are distinct, and q queries. In each query, you will given two integers l and r, and you need to determine $f(a[l \ldots r])$ where $a[l \ldots r]$ is the subarray‡ of a starting at index l and ending at index r. Additionally, it is guaranteed that this subarray is a balanced sequence.

*A sequence c is a subsequence of sequence b if c can be obtained from b by deleting some (possibly zero or all) elements.

†A partition of a sequence b is a way to divide it into one or more subsequences such that every element of b belongs to exactly one subsequence. In other words, each element must be used exactly once across all subsequences in the partition.

‡A sequence a is a subarray of a sequence b if a can be obtained from b by deletion of several (possibly, zero or all) elements from the beginning and several (possibly, zero or all) elements from the end.

Input

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

The first line of each test case contains two integers n and q ($1 \le n, q \le 10^5$) — the length of the sequence a and the number of queries.

The second line contains $2n$ integers $a_1, a_2, \ldots, a_{2n}$ ($1 \le |a_i| \le n, a_i \neq 0$). **It is guaranteed that a is a balanced sequence and all elements of a are distinct.**

Each of the next q lines contains two integers l and r ($1 \le l < r \le 2n$) representing the indices of the subarray $a[l \ldots r]$. **It is guaranteed that $a[l \ldots r]$ is also a balanced sequence.**

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

Output

For each query, print a single integer — the minimum number of subsequences into which the specified subarray can be partitioned into such that each subsequence is a primal sequence.

Example

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