Problem A. Edgy Graph

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

You are given an undirected graph with n vertices and m edges, where each edge has a positive weight.

Your task is to construct an array a of size n such that the following conditions are satisfied:

- $1 \le a_i \le 10^9$ for all $1 \le i \le n$
- for every edge (u, v) between vertices u and v having weight w, $\max(a_u, a_v) = w$. That is, the maximum value of the two vertices of the edge must be equal to the weight of the edge.

Input

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

The first line of each test case contains two integers n and m ($2 \le n \le 3 \cdot 10^5$, $1 \le m \le 3 \cdot 10^5$) — the number of vertices and the number of edges in the graph.

The next m lines describe the edges. Each line contains three integers u, v, and w ($1 \le u, v \le n$, $1 \le w \le 10^9$) — the vertices connected by the edge and its weight. It is guaranteed that the graph has no self-loops or multiple edges.

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

Output

For each test case, output a single line containing n integers $a_1, a_2, \ldots, a_n$ — the array a that satisfies the given conditions. If there are multiple such arrays, output any of them. If no such array exists, output -1 instead.

Example

standard input	standard output
3 3 3 1 2 1 2 3 2 3 1 2 3 3 1 2 1 2 3 2 1 3 3 3 1 1 2 5	1 1 2 -1 1 5 1000000000

Problem B. Red Dead Redemption 2

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

After a successful train robbery, outlaw Dutch and his gang acquired n valuable items with values $a_1, a_2, \ldots, a_n$. An old-timer warned them that items can only be stored together if their values are pairwise coprime (that is, every pair of items shares no common factors other than 1), or bad luck will follow and the gang will be dead.

Arthur, Dutch's most trusted gang member, has been tasked with storing the loot. For security reasons, he must split the loot between two hideouts such that both hideouts contain at least one item, and each item is stored in exactly one hideout.

Arthur now wants to know how many different ways he can distribute the stolen goods so that the items in each hideout can be safely stored together. Two distributions are considered different if there exists at least one item placed in different hideouts (each item is identified by its index, not value).

As the count can be large, Arthur needs to calculate it modulo 998244353.

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 an integer n ($2 \le n \le 5 \cdot 10^5$) — the number of items.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 5 \cdot 10^6$) — the values of the items.

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

Output

For each test case, output a single integer — the number of safe distributions modulo 998244353.

Example

standard input	standard output
1 5 33 21 70 55 52	2

Note

One valid split of items in the given case is $\{1, 3\}$ and $\{2, 4, 5\}$ (where these numbers represent indices). Swapping the groups is also valid.

Problem C. Binomial XOR

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

Let $M = 998244353$.

You are given an integer n. For an integer i $(1 \le i \le n)$, define $f(i)$ as follows:

$$f(i) = \left(\binom{i}{i} \bmod M\right) \oplus \left(\binom{i+1}{i} \bmod M\right) \oplus \cdots \oplus \left(\binom{n}{i} \bmod M\right)$$

Formally,

$$f(i) = \bigoplus_{j=i}^{n} \left(\binom{j}{i} \bmod M\right)$$

Here, $\binom{x}{y}$ denotes the binomial coefficient, which represents the number of ways to choose y items from a set of x items, and $\oplus$ denotes the bitwise XOR operation.

Note that to compute $f(i)$, you first compute the modulo operation on each binomial coefficient, then apply the XOR operation. You should **not** take $f(i) \bmod M$.

Please output $f(1) \oplus f(2) \oplus \cdots \oplus f(n)$.

Input

The first line contains a single integer t $(1 \le t \le 10^6)$ — the number of test cases.

The first and only line of each test case contains a single integer n $(1 \le n \le 10^6)$.

Output

For each test case, output the answer.

Example

standard input	standard output
2 2 1	2 1

Note

In the first test case,

- $f(1) = \left(\binom{1}{1} \bmod M\right) \oplus \left(\binom{2}{1} \bmod M\right) = 1 \oplus 2 = 3$
- $f(2) = \left(\binom{2}{2} \bmod M\right) = 1$

So the answer is $f(1) \oplus f(2) = 3 \oplus 1 = 2$.

In the second test case,

- $f(1) = \left(\binom{1}{1} \bmod M\right) = 1$

So the answer is $f(1) = 1$.

Problem D. Symmetric Swap

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4.5 seconds
Memory limit:	256 megabytes

For an array of integers $c_1, c_2, \ldots, c_n$, you can transform it by performing the following operation any number of times (including zero):

- Select an integer ℓ such that $1 \leq \ell \leq \lfloor \frac{n}{2} \rfloor$. Then, swap the prefix of length ℓ with the suffix of length ℓ.

For example, if $c = [1, 2, 3, 4, 5, 6]$ of length 6, you can select ℓ such that $1 \leq \ell \leq \lfloor \frac{6}{2} \rfloor = 3$. If you select $\ell = 2$, you swap the prefix $[1, 2]$ with the suffix $[5, 6]$, resulting in $[5, 6, 3, 4, 1, 2]$.

Let $f(c, i)$ denote the i-th left cyclic shift of array c. That is, $f(c, i) = [c_{i+1}, c_{i+2}, \ldots, c_n, c_1, c_2, \ldots, c_i]$. For example, if $c = [1, 2, 3, 4, 5, 6]$, then $f(c, 2) = [3, 4, 5, 6, 1, 2]$.

You are given two arrays a and b, both of length n. Your task is to find the number of pairs of integers (i, j) where $0 \leq i, j < n$ such that $f(a, i)$ can be transformed into $f(b, j)$ using the aforementioned operations.

Input

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

The first line of each test case contains a single integer n ($1 \leq n \leq 3 \cdot 10^5$) — the length of both arrays.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^6$) — the elements of array a.

The third line contains n integers $b_1, b_2, \ldots, b_n$ ($1 \leq b_i \leq 10^6$) — the elements of array b.

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 a single integer — the number of pairs of integers (i, j) where $0 \leq i, j < n$ such that $f(a, i)$ can be transformed into $f(b, j)$.

Example

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

Note

In the first test case, as $n = 3$, so you can select ℓ such that $1 \leq \ell \leq \lfloor \frac{3}{2} \rfloor = 1$. It can be shown that there are 3 valid pairs for this case. For example, $(1, 1)$ is a valid pair, as $f(a, 1) = [2, 3, 1]$ can be transformed to $f(b, 1) = [1, 3, 2]$ by swapping the prefix $[2]$ with the suffix $[1]$.

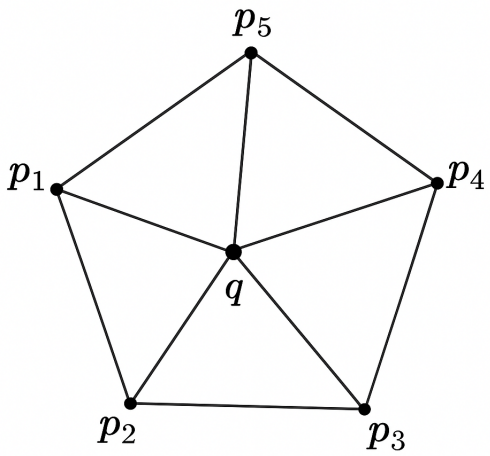
Problem E. The Perfect Spider Web

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

Disaster has struck New York City! The villainous Green Goblin has launched a devastating attack on n skyscrapers positioned at coordinates $p_1, p_2, \ldots, p_n$. The buildings are arranged in counter-clockwise order, and no three buildings are collinear. The coordinates form a convex polygon, which is called the city block.

The Goblin's pumpkin bombs have severely damaged the buildings, and they are about to lean dangerously outward away from the center of the block. Spider-Man must act quickly! He needs to position himself at a point q **strictly inside** the city block and shoot webs directly to each building to prevent them from falling and keep them balanced.

When Spider-Man shoots webs from position q to all n buildings, he creates n triangular tension zones $\triangle p_i\, p_{i+1}\, q$ for $i = 1, 2, \ldots, n$ (where $p_{n+1} = p_1$). Each zone represents the structural stress distribution area that his web must support.



The larger a triangular tension zone, the more web fluid is required. With limited web fluid, Spider-Man must choose a position q to minimize the maximum tension zone area to ensure he can stabilize all buildings.

Help Spider-Man save the city by finding the optimal position that minimizes the maximum triangle area! Output this minimum possible maximum area.

Input

The first line contains an integer n ($3 \le n \le 500$) — the number of damaged buildings surrounding the city block.

Each of the following n lines contains two space-separated integers x_i and y_i ($1 \le x_i, y_i \le 10^6$) — the coordinates of the i-th building's position.

It is guaranteed that the buildings form a convex polygon in counter-clockwise order and no three buildings are collinear.

Output

Output a single real number — the minimum possible maximum triangle area that Spider-Man can achieve with optimal positioning.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}. Formally, if your answer is a and the jury's answer is b, your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Example

standard input	standard output
4 1 1 2 1 2 2 1 2	0.2500000000

Note

One possible choice is $q = (1.5, 1.5)$. Then the area of each triangle is 0.25, so the maximum of these four triangle areas is 0.25. It can be shown that no other choice of q can make this maximum smaller.

Problem F. Divisible Perfection

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

You are given a digit string s of length n where each character is a digit from 1 to 9.

You need to determine if, for each pair of indices $1 \leq i \leq j \leq n$, the number formed by the contiguous substring $s_i s_{i+1} \cdots s_j$ is divisible by its length $(j - i + 1)$.

Input

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

The first line of each test case contains a single integer n $(1 \leq n \leq 3 \cdot 10^5)$ — the length of the string.

The second line contains a string s of length n consisting of digits from 1 to 9.

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 `YES` if every substring is divisible by its length, and `NO` otherwise.

Example

standard input	standard output
3 3 162 2 69 1 7	YES NO YES

Note

In the first test case, all substrings are divisible by their lengths:

- All 1 length substrings 1, 6 and 2 are divisible by 1.
- All 2 length substrings 16 and 62 are divisible by 2.
- The 3 length substring 162 is divisible by 3.

So the answer is `YES` for this test case.

In the second test case, the substring 69 is not divisible by its length 2, so the answer is `NO` for this test case.

Problem G. MEX-imum Beauty

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

For an array b of length m, define the beauty of b as the MEX* of the sequence of prefix maximums of b. Formally,

$$\text{MEX}(\{\max(b_1), \max(b_1, b_2), \max(b_1, b_2, b_3), \ldots, \max(b_1, b_2, \ldots, b_m)\})$$

Given an array a of n integers, please compute the sum of the beauty of all non-empty subarrays† of a.

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

† A sequence b is a subarray of a sequence a if b can be obtained from a 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 a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains an integer n ($1 \le n \le 10^6$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le n$).

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output a single integer — the sum of the beauty of all non-empty subarrays of a.

Example

standard input	standard output
4 6 3 0 1 0 2 4 5 0 1 2 3 4 6 2 0 3 5 1 0 3 0 3 1	14 15 6 3

Note

In the first test case, the sum of the beauty of all non-empty subarrays is 14. In particular, the subarray from index 2 to 5 is $[0, 1, 0, 2]$. Its beauty is $\text{MEX}(\{\max(0), \max(0, 1), \max(0, 1, 0), \max(0, 1, 0, 2)\}) = \text{MEX}(\{0, 1, 1, 2\}) = 3$.

Problem H. Substring Symphony

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

You are given two strings a and b, where a has length n and b has length m.

For a string c, define $f(c)$ as the number of distinct integers k $(1 \le k \le |c|)$ such that every k-length contiguous substring of c is also a contiguous substring of a.

Your task is to answer q queries. In each query, you are given two integers l and r, and you need to compute $f(b[l : r])$, where $b[l : r]$ denotes the contiguous substring of b starting at index l and ending at index r.

Input

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

The first line of each test case contains three integers n, m, and q $(1 \le n, m \le 4 \cdot 10^5, 1 \le q \le 10^6)$ — the lengths of strings a and b, and the number of queries.

The second line contains the string a of length n consisting of lowercase English letters.

The third line contains the string b of length m consisting of lowercase English letters.

Each of the next q lines contains two integers l and r $(1 \le l \le r \le m)$.

It is guaranteed that the sum of n and the sum of m over all test cases does not exceed $4 \cdot 10^5$, and the sum of q over all test cases does not exceed 10^6.

Output

For each query, output a single integer — the value of $f(b[l : r])$.

Example

standard input	standard output
2 5 5 3 abdbc cabce 2 4 5 5 3 4 2 3 1 aa abc 1 1	2 0 2 1

Note

In the first test case, for the first query, we have $a =$ `abdbc` and $c = b[2 : 4] =$ `abc`. Here,

- For $k = 1$, the 1-length substrings of c are `a`, `b`, `c`. All of these appear in a, so $k = 1$ is valid.
- For $k = 2$, the 2-length substrings of c are `ab`, `bc`. Both appear in a, so $k = 2$ is valid.
- For $k = 3$, the 3-length substring of c is `abc`. This does not appear in a, so $k = 3$ is not valid.

Therefore, $f(c) = 2$ as for two different values of k, the substrings of c appear in a.

Problem I. Statue on a Permutation

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

Alice and Bob are playing a game on a permutation p of length n, where p contains each integer from 1 to n exactly once.

The game is played with a statue that can be placed on any position of the permutation. Initially, the statue is placed at position x. Alice moves first, and the players alternate turns.

On each turn, the current player can move the statue from its current position i to any position j (where $i \neq j$), but only if $p_i \geq p_k$ for all indices k such that $\min(i, j) \leq k \leq \max(i, j)$.

The first player who cannot make any valid move loses the game.

Both Alice and Bob play optimally, meaning that at their turns they perform the best possible move. Due to this, the game becomes quite deterministic if you know the initial permutation p and the initial position of the statue, x. In fact, for a given permutation p, we can figure out for each initial position x whether Alice or Bob will win the game.

You are given a string s of length n consisting of letters `A` and `B`. Your task is to construct a permutation p of length n such that, for each position x, if the game starts with the statue at position x, the winner is Alice if the x-th character of the string is `A`, and Bob if it is `B`.

Input

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

The first line of each test case contains an integer n ($1 \leq n \leq 1000$) — the length of the string.

The second line contains a string s of length n consisting only of characters `A` and `B`.

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

Output

For each test case, output a single line containing n integers $p_1, p_2, \ldots, p_n$ — the permutation p that satisfies the conditions.

If there are multiple possible permutations, output any one of them. If no such permutation exists, output -1 instead.

Example

standard input	standard output
3 3 BAB 3 AAA 4 BAAA	1 3 2 -1 1 2 3 4

Problem J. Sublime Replacement

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

A non-decreasing subarray of an array a of length n is a contiguous segment $a_i, a_{i+1}, \ldots, a_j$ where $1 \le i \le j \le n$ and $a_i \le a_{i+1} \le \cdots \le a_j$.

The score of an array a is defined as the maximum of $a_i + a_{i+1} + \cdots + a_j$ over all non-decreasing subarrays of a.

You are given an array a of n integers, where some elements are equal to -1. Your task is to replace each -1 with a positive integer from the range $[1, 10^9]$ (inclusive) such that the score of the resulting array is maximized.

Input

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

The first line of each test case contains a single integer n ($1 \le n \le 3 \cdot 10^5$) — the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$ or $a_i = -1$).

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 a single integer — the maximum possible score after optimally replacing all -1 values.

Example

standard input	standard output
3 4 1 3 2 4 5 4 3 -1 -1 2 1 1	6 2000000003 1

Note

In the first test case, the best subarray is from index 3 to 4, which is $[2, 4]$, and it has a sum of 6. This is the maximum score over all non-decreasing subarrays of the array.

In the second test case, we can replace both -1 with 10^9, resulting in $[4, 3, 10^9, 10^9, 2]$. The best subarray is from index 2 to 4, which is $[3, 10^9, 10^9]$, and it has a sum of 2000000003. This is the maximum possible score after replacing all -1 values.

Problem K. Math Madness

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Given a sequence of n integers $a_1, a_2, \ldots, a_n$, please count the number of pairs of indices (i, j) such that $1 \le i \le j \le n$ and $\frac{\text{lcm}(a_i, a_j)}{\gcd(a_i, a_j)}$ is divisible by $\max(a_i, a_j)$.

Here, $\text{lcm}(x, y)$ denotes the least common multiple of x and y, and $\gcd(x, y)$ denotes the greatest common divisor of x and y.

Input

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

The first line of each test case contains an integer n ($1 \le n \le 10^6$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10 \cdot n$).

It is guaranteed that the sum of n over all the test cases does not exceed 10^6.

Output

For each test case, output a single integer — the number of pairs of indices that satisfy the condition.

Example

standard input	standard output
4 3 2 3 3 5 2 3 2 2 7 3 3 3 3 6 2 10 23 12 10 40	2 7 0 5