Problem A. Ascending Subarray

Input file:	standard input
Output file:	standard output
Balloon Color:	Orange

You are given an array a of n elements. This array has two pointers, and you will be given the initial position of these pointers.

Your task is to move these two pointers so that when you reverse the subarray between them inclusive, the subarray will be sorted **ascendingly**.

Print the maximum length of this subarray and the minimum number of operations required to move those two pointers towards it.

Input

The input consists of multiple test cases.

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 three integers n, l, r $(1 \leq l \leq r \leq n \leq 10^5)$ — length of the array, left pointer, right pointer.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq 10^9)$ — elements of the array a.

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

Output

For each test case, print two integers — the maximum length of this subarray, the minimum number of operations required to move those two pointers towards the subarray.

Example

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

Problem B. Game

Input file: `standard input`
Output file: `standard output`
Balloon Color: `White`

Alaa and Habiba played a game where Alaa scored A and Habiba scored H.

Can you determine who won or if it was a draw?

Input

One line contains two integers number A, H $(1 \le A, H \le 100)$.

Output

Print "A"if Alaa won, "H"if Habiba won, or "D"if it was a draw.

Example

standard input	standard output
2 3	H

Problem C. Great Coaches

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Yellow`

Amr and Hazem are excellent problem-solving coaches. Their trainees wanted more challenges, specifically involving arrays. To meet this demand, the coaches presented an original problem that the trainees had never encountered before. The problem is as follows:

Given an array a of length n, count the number of subsequences of this array such that the Least Common Multiple (LCM) of the elements in the subsequence is equal to x. Since the number of such subsequences can be very large, print the count modulo $10^9 + 7$.

A subsequence is a sequence that can be derived from another sequence by deleting zero or more elements without changing the order of the remaining elements.

The Least Common Multiple (LCM) of a set of integers is the smallest positive integer that is divisible by each of the integers in the set.

Input

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

For each test case:

The first line contains two integers n $(1 \le n \le 10^5)$ and x $(1 \le x \le 10^7)$ — the number of elements in a and the LCM, respectively.

The second line contains n numbers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^7)$ — the elements of a.

It's guaranteed that the sum of n over all test cases doesn't exceed 10^5.

Output

You should output one number - the number of subsequences that their $LCM = x$ modulo $10^9 + 7$.

Example

standard input	standard output
4 5 10 2 5 3 10 2 4 7 7 7 7 7 7 3 1 2 3 4 5 6 7 10 10 2 4 6 3 4 8 7 14 15 10	11 15 2 2

Problem D. Waheed's Array

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Silver`

Waheed has an array of integers. He would love to know how many operations it will take to transform his array into a palindrome array.

The operation is as follows: he can take any element from the array and divide it by one of its divisors. Waheed can do the operation any number of times (possibly zero).

Can you tell the minimum number of operations required to transform the array?

Input

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

The first line of each test case contains one integer n, the size of the array$(1 \le n \le 10^5)$.

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 overall test cases does not exceed 10^5.

Output

For each test case, output a single number: the minimum number of operations on a separate line.

Example

standard input	standard output
2 4 8 5 3 12 3 1 1 1	4 0

Problem E. Common Subsequences

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Red`

Given an array of size n, an integer x, and q queries.

Each query consists of two integers l and r. For each query, determine the number of subsequences that can be formed from the array elements between indices l and r, such that the subsequence has x as a common divisor.

The result for each query should be computed modulo $10^9 + 7$.

Input

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

Then follows T test cases.

The first line of each test case contains n, x and q $(1 \leq n, q \leq 10^5)$ $(1 \leq x \leq 10^9)$ — the size of the array, the common divisor, and the number of queries respectively.

The second line of each test case contains n elements $(1 \leq a_i \leq 10^9)$ — the elements of the array.

Then q queries, each query contains l, r $(1 \leq l \leq r \leq n)$ — indices of the range.

It is guaranteed that the sum of n and q overall test cases does not exceed 10^5.

Output

Output the answer of the queries for each test case modulo $10^9 + 7$ on a separate line.

Example

standard input	standard output
3 4 2 3 2 3 4 8 1 4 2 3 3 3 6 3 3 2 3 4 8 9 12 2 5 1 6 6 6 10 1 3 2 10 7 1 5 4 8 9 12 1 3 7 2 4 5 8	7 1 1 3 7 1 31 7 15

Problem F. Split the subarray

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Blue`

Given an array of n integers, you can choose any position p such that $(p > 1)$, and calculate the maximum subarray sum for the elements to the left of that position and the maximum subarray sum for the elements from that position to the right.

In other words, if you choose position p (1-based index and $p > 1$), you need to find the maximum subarray sum from $(a_1, a_2, \ldots, a_{p-1})$, and the maximum subarray sum from $(a_p, a_{p+1}, \ldots, a_n)$.

Your task is to minimize the absolute difference between these two sums by choosing the optimal position p. Output the minimum possible difference.

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 a single integer n $(2 \le n \le 10^5)$ — the length of the array.

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

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 one integer — the minimum possible difference.

Example

standard input	standard output
2 5 1 -3 4 2 1 3 -2 2 2	1 0

Problem G. Replace From Range

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Rose`

Given an integer array a of size N.

If $a_i = -1$, that means you can replace it with any other number from the range $[L, R]$.

Count the number of ways you can replace all indices i where $a_i = -1$ such that no two consecutive numbers are equal.

It's guaranteed that $a_1 \neq -1$ and $a_n \neq -1$.

As the number can be very big, print the answer modulo $10^9 + 7$.

Input

The input consists of multiple lines.

The first line contains three integers N, L, R $(1 \leq N \leq 10^5, 1 \leq L \leq R \leq 10^{18})-$ the size of the string, and the range of numbers that you can choose from.

The second line contains N integers $a_1, a_2, a_3, ..., a_N$ $(L \leq a_i \leq R$, or $a_i = -1)$.

Output

Single integer C – the number of ways you can replace all indices i where $a_i = -1$ such that no two consecutive numbers are equal.

As the number can be very big, print C modulo $10^9 + 7$.

Examples

standard input	standard output
4 1 5 5 -1 -1 5	12
4 6 10 7 -1 -1 6	13

Problem H. Jana and the Tournament

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Black`

Jana loves tournaments. She is given an integer n, the number of participants in a tournament, numbered from 1 to n. The tournament is structured in multiple rounds. In each round, matches are held between pairs of players as follows:

- Player 1 plays against Player 2
- Player 3 plays against Player 4
- And so on...

The winner from match 1 plays the winner from match 2, the winner of match 3 plays the winner of match 4, etc., in the subsequent round. The winner of a match is the participant whose largest divisor (excluding the number itself) is the biggest. In case of a tie, the bigger number wins. If a participant does not have someone to play against in a particular round, they advance to the next round automatically.

Jana wants to find out who will win the tournament.

Player 1's largest divisor is considered equal to 1.

Input

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

The onlyline contains an integer n $(1 \le n \le 10^6)$, the number of participants.

Output

For each test case, output a single integer: the number of the participant who won the tournament on a separate line.

Example

standard input	standard output
3 1 15 16	1 14 16

Problem I. Shahd and Descending Numbers

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Green`

Shahd loves math and numbers. She has a special fondness for certain numbers based on their binary representations. One day, she discovered something interesting: some numbers have binary representations where the bits form a descending sequence. For example, the numbers 1, 4, and 6 have this property, while 5 and 9 do not. Shahd refers to these numbers as "descending numbers."

Fascinated by this discovery, Shahd wants to find out how many descending numbers exist within a given range. She asks for your help to count the number of descending numbers between l and r (inclusive).

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 l and r ($1 \le l \le r \le 10^9$).

Output

For each test case, print the number of descending numbers in the range $[l, r]$ inclusive.

Example

standard input	standard output
3 4 6 12 15 8 9	2 3 1

Note

In the first test case where $l = 4$ and $r = 6$

1. **Numbers in range and their binary representations:**
   - 4 - 100 descending
   - 5 - 101 not descending
   - 6 - 110 descending
2. **Count of descending numbers:** 2

Problem J. Sherlock and Equal Arrays

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Purple`

Mariam loves Sherlock Holmes and aspires to be his student to learn from his great experience. To see if she is worthy of his time, Sherlock gave her an easy challenge.

Given two arrays A and B, the goal is to make array A equal to B using the following types of operations:

1. Choose an integer x $(0 \le x < 30)$ and a subset of elements from A, and apply the bitwise OR operation to each element in the subset with 2^x.
2. Choose an integer x $(0 \le x < 30)$ and a subset of elements from B, and apply the bitwise OR operation to each element in the subset with 2^x.

Mariam needs your help to determine the minimum number of operations (possibly zero) required to make arrays A and B equal.

Input

The first line contains a single integer t $(1 \le t \le 10^3)$ — the number of test cases.

Each test case has 3 lines:

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

The second line contains n integers $A_1, A_2, \ldots, A_n$ $(0 \le A_i < 2^{30})$ — elements of the array A.

The third line contains n integers $B_1, B_2, \ldots, B_n$ $(0 \le B_i < 2^{30})$ — elements of the array B.

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

Output

For each test case, print one integer — the minimum number of operations (possibly zero) to make arrays A and B equal.

Example

standard input	standard output
3 1 1 1 4 7 4 6 2 1 5 4 3 5 16 7 3 9 5 10 1 2 4 3	0 3 8

Note

Two arrays A and B of length n are considered equal if $A_i = B_i$ for all $i, (1 \le i \le n)$.

Problem K. Rady the farmer

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Light Blue`

Rady, a lazy farmer, wants to maximize his profit with minimal effort. Here's his detailed plan:

Each day, Rady performs the following operations:

- He triples the fruit count of each existing tree.
- He plants y new trees, with each new tree initially having 1 fruit.

Rady will be satisfied if he manages to collect at least z fruits in total.

Initially, Rady starts with x trees, each initially having 1 fruit. Calculate the minimum number of days Rady needs to work to achieve satisfaction.

Input

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

The first line of each test contains three integers x, y, z $(1 \le x, y, z \le 10^{18})$ — the number of trees he will start with, the number of trees he will plant each day, and the number of fruits that will make him satisfied.

Output

For each test case, output one integer — the minimum number of days that Rady needs to work to be satisfied.

Example

standard input	standard output
3 1 1 1 3 4 4 2 3 9	0 1 1

Problem L. Young Siblings

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Bronze`

Reem introduced a two-player game to her younger siblings, Toty and Boty. Here's how the game works:

There are n cards, each labeled with a number a_i where $1 \le i \le n$. Toty always starts first. Players take turns picking cards either from the beginning or the end of the sequence until no cards are left. Each player aims to maximize their score, defined as the bitwise XOR of all the cards they picked. The player with the higher score wins, and if both play optimally, determine whether Toty wins, Boty wins, or if it results in a draw.

Input

The first line contains an integer n ($1 \le n \le 10^2$) - the number of cards.

The second line contains n integers - a_i ($1 \le a_i \le 10^3$) representing the values on the cards.

Output

If Toty wins, output "`Toty`". If Boty wins, output "`Boty`". If it's a draw, output '`Toty-Boty`"

Examples

standard input	standard output
4 1 2 3 4	Toty
4 2 2 2 2	Toty-Boty
3 1 3 1	Boty

Problem M. Fibonacci's Store

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Light Green`

Gamal buys groceries on Fibonacci days, which correspond to Fibonacci numbers. He buys the same items on two different Fibonacci days n and m $(n > m)$ if m divides n.

Given n, the n-th Fibonacci number is denoted as F_n. Gamal wants to count how many Fibonacci days he bought the same items as F_n, including F_n itself.

To clarify:

- Fibonacci days are those corresponding to Fibonacci numbers.
- Fibonacci numbers are defined recursively: $F_1 = 1, F_2 = 1, F_n = F_{n-1} + F_{n-2}$ for $n > 2$.
- Gamal buys the same items on two Fibonacci days n and m $(n > m)$ if m divides n.

Note: F_1 and F_2 are considered as two different days.

Input

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

Then follows t test cases. Each test case contains an integer n $(1 \leq n \leq 10^7)$.

Output

For each test case, output a single integer representing the number of Fibonacci days Gamal bought the same items as F_n, on a separate line.

Example

standard input	standard output
6 1 2 4 8 9 12	2 2 3 4 4 6

Note

The first 4 Fibonacci numbers are $1, 1, 2, 3$.

- For the first test case where $n = 1$, $F_1 = 1$, there are 2 days m that divide F_1: $F_1 = 1$ and $F_2 = 1$.
- For the second test case where $n = 2$, $F_2 = 1$, there are 2 days m that divide F_2: $F_1 = 1$ and $F_2 = 1$.
- For the third test case where $n = 4$, $F_4 = 3$, there are 4 days m that divide F_4: $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, and $F_4 = 3$.