Good Intervals (`intervals`)

For some $k \geq 1$, we say that the sequence of integers $v = [v_1, v_2, \ldots v_k]$ is **good** if v_i is divisible by i for every i from 1 to k.

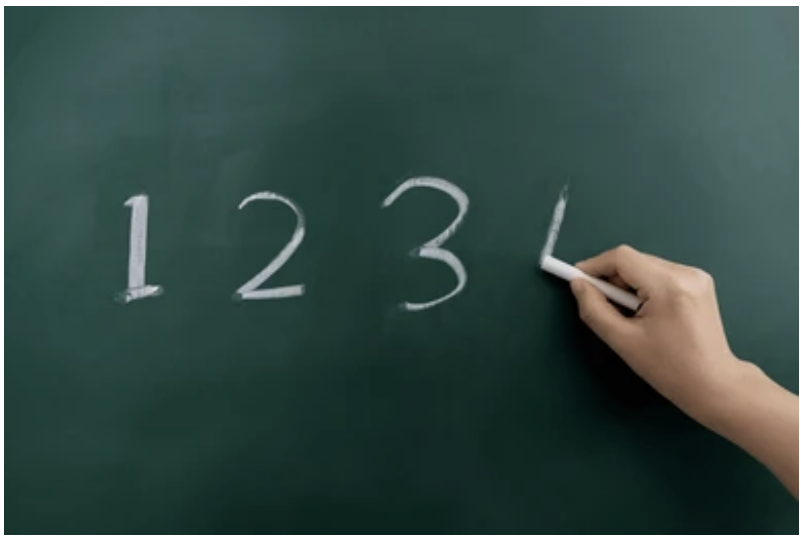

Figure 1: Writing good sequences is a fashionable pastime!

You are given an integer sequence $A = [A_1, A_2, \ldots, A_N]$ of length N and Q queries of the form l_i, r_i, which represent the range of values $[A_{l_i}, A_{l_i+1}, \ldots, A_{r_i}]$ from the sequence. For each query compute the number of subintervals of the given range which are **good** sequences (each subinterval is considered as an independent sequence, indexed from 1).

> ☞ Among the attachments of this task you may find a template file `intervals.*` with a sample incomplete implementation.

Input

The first line of the input file contains a single integer T, the number of test cases. T test cases follow, each preceded by an empty line.

The first line of each test case contains a single integer N, the length of the sequence.

The second line of each test case contains N space-separated integers A_i, the elements of the sequence.

The third line of each test case contains a single integer Q, the number of queries.

Each of the following Q lines contain two integers l_i, r_i, the query intervals.

Output

For each test case, output all query results, each on a separate line.

Constraints

- $1 \leq T \leq 10$.
- $1 \leq N, Q \leq 100\,000$.
- $1 \leq A_i \leq 10^{18}$.
- $1 \leq l_i \leq r_i \leq N$.
- The sum of N and Q over all test cases does not exceed $100\,000$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (10 points) $N, Q \leq 200$.

– **Subtask 3** (20 points) $N \leq 2000$.

– **Subtask 4** (30 points) $A_i \leq 10^6$ for every $i = 1 \ldots N$.

– **Subtask 5** (40 points) No additional limitations.

Examples

input	output
1 5 6 24 6 8 10 3 1 3 2 4 1 5	6 5 12

Explanation

In the **sample case**, consider the the second query corresponding to the range of values $[24, 6, 8]$.

- An example of a good subinterval is $[A_2, A_3] = [24, 6]$, because 24 is divisible by 1 and 6 is divisible by 2.
- An example of a subinterval which is not good is $[A_2, A_3, A_4] = [24, 6, 8]$, as 8 is not divisible by 3.

The 5 good subintervals for this query are the sequences $[24]$, $[24, 6]$, $[6]$, $[6, 8]$ and $[8]$.