BINUS University

Academic Career: ***Undergraduate / ~~Master~~ / ~~Doctoral~~ *)***	**Class Program:** ***~~International~~ / Regular / ~~Smart Program~~ /*** ***Global Class / ~~BINUS Online Learning~~ *)***
☑ **Mid Exam** ☐ **Compact Term Exam** ☐ **Final Exam** ☐ **Others Exam : _____**	**Term : Odd / ~~Even~~ / ~~Compact~~ *)** **Period (Only for BOL) : ~~1~~ / ~~2~~ *)**
☑ **Kemanggisan** ☐ **Senayan** ☐ **Semarang** ☑ **Alam Sutera** ☐ **Bandung** ☑ **Bekasi** ☐ **Malang**	**Academic Year :** **2022 / 2023**

Exam Type*	:	Onsite / Online	Faculty / Dept.	:	School of Computer Science
Day / Date**	:		Code - Course	:	COMP6226001 – Competitive Programming
Time**	:		Code - Lecturer	:	Team Teaching
Exam Specification***	:	☐ Open Book ☐ Open Notes ☐ Close Book ☐ Submit Project ☐ Open E-Book ☐ Oral Test	BULC (Only for BOL)	:	-
			Class	:	
Equipment***	:		Student ID ***	:	
☐ Exam Booklet ☐ Laptop ☐ Drawing Paper – A3			Name ***	:	
☐ Calculator ☐ Tablet ☐ Drawing Paper – A2			Signature ***	:	
☐ Dictionary ☐ Smartphone ☐ Notes					

*) *Strikethrough the unnecessary items* **) *For Online Exam, this is the due date* ***) *Only for Onsite Exam*

Please insert the test paper into the exam booklet and submit both papers after the test.

The penalty for CHEATING is DROP OUT!

Learning Outcomes:

LO1: (C3) Application : apply algorithm techniques and methods
LO2: (C4) Analysis : calculate processing time and memory space of algorithms.
LO3: (C5) Synthesis : Create good and correct algorithm for problem solving

I. Case Study (100%)

Please submit your answer in the provided online judge. Your score will be taken from the online judge.

Verified by, *[Lecturer Name] (Lecturer ID) and sent to Department/Program on MMM DD, YYYY*

A. Jojo and GCD Subsequence (30 points)

You are given N integers, $A_1, A_2, \ldots, A_N$.
Denote the *GCD* of a subsequence as the greatest common divisor of all elements in the subsequence.
For each i that satisfy $1 \leq i \leq N$, determine the maximum GCD of all subsequences that consist of exactly i elements.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq A_i \leq 10^6$, for $1 \leq i \leq N$

Subtasks

1. (10 points) $N = 3$
2. (10 points) $1 \leq A_i \leq 100$, for $1 \leq i \leq N$
3. (10 points) No additional constraints.

Input Format

N $A_1\ A_2\ \ldots A_N$

Output Format

Output N integers in a single line. The i-th integer represents the maximum GCD of all subsequences that consist of exactly i elements, for each i that satisfy $1 \leq i \leq N$.

Sample

Input	*Output*
4 2 4 6 3	6 3 2 1
3 4 4 4	4 4 4
1 10	10

Explanation

For the first sample, the maximum GCD of subsequences that consist of 1, 2, 3, and 4 elements are $\{6\}$, $\{6, 3\}$, $\{2, 4, 6\}$, and $\{2, 4, 6, 3\}$, respectively.

For the second sample, all subsequences have a GCD of 4.

Verified by, *[Lecturer Name] (Lecturer ID) and sent to Department/Program on MMM DD, YYYY*

B. Lili and Range Query (40 points)

You are given N integers, $A_1, A_2, \ldots, A_N$.
Denote the *score* of an array B as the sum of the product of all elements and its index. For instance, the score of an array $B = [4,5,1]$ is $4 \cdot 1 + 5 \cdot 2 + 1 \cdot 3 = 17$.
You want to answer Q queries (numbered from 1 to Q). For query i, you are given two integers L_i and R_i. First, convert the subarray $A_{L_i \ldots R_i}$ into a new array, which index starts from 1. Then, determine the score of the new array.

Constraints

- $1 \leq N, Q \leq 10^5$
- $1 \leq A_i \leq 10^9$, for $1 \leq i \leq N$
- $1 \leq L_i \leq R_i \leq N$, for $1 \leq i \leq Q$

Subtasks

1. (10 points) $L_i = R_i$, for $1 \leq i \leq Q$
2. (10 points) $1 \leq N, Q \leq 100$
3. (10 points) $L_i = 1$, for $1 \leq i \leq Q$
4. (10 points) No additional constraints.

Input Format

$N\ Q$ $A_1\ A_2 \ldots A_N$ $L_1\ R_1$ $L_2\ R_2$ ... $L_Q\ R_Q$

Output Format

For each query, determine the score of the new array in that query.

Sample

Input	*Output*
4 3 2 4 6 3 2 4 3 3 1 4	25 6 40

Explanation

For the first sample:

- In the first query, the score of subarray $[4, 6, 3]$ is $4 \cdot 1 + 6 \cdot 2 + 3 \cdot 3 = 25$
- In the second query, the score of subarray $[6]$ is $6 \cdot 1 = 6$
- In the third query, the score of subarray $[2, 4, 6, 3]$ is $2 \cdot 1 + 4 \cdot 2 + 6 \cdot 3 + 3 \cdot 4 = 40$

Verified by, *[Lecturer Name] (Lecturer ID) and sent to Department/Program on MMM DD, YYYY*

C. Bibi and Homework (30 points)

You have N homeworks (numbered from 1 to N) for this semester.
Homework i appears on day T_i. You can only solve a homework if it already appears, i.e., the current day is not less than T_i. Initially, you are at day 0.
Homework i requires L_i days to finish. Formally, if you start working on day x, then you will finish on day $x + L_i - 1$. For each day, you can only work on **one** homework. Therefore, during day x until $x + L_i - 1$ (inclusive), you can only work on homework i.
Determine the fastest day such that all homeworks are finished.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq T_i, L_i \leq 10^9$, for $1 \leq i \leq N$

Subtasks

1. (10 points) $T_i = 1$, for $1 \leq i \leq N$
2. (10 points) $L_i = 1$, for $1 \leq i \leq N$
3. (10 points) No additional constraints.

Input Format

N $T_1\ L_1$ $T_2\ L_2$ ... $T_N\ L_N$

Output Format

Output an integer which represents the fastest day such that all homeworks are finished.

Sample

Input	*Output*
3 1 3 1 1 1 5	9
3 2 1 5 1 3 1	5
3 1 3 2 5 4 1	9

Explanation

For the third sample, you can start working on homework 1 on day 1, homework 3 on day 4, and homework 2 on day 5.

Verified by, *[Lecturer Name] (Lecturer ID) and sent to Department/Program on MMM DD, YYYY*