

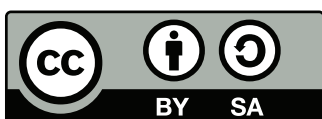
GRCPC 2025

Greek Collegiate Programming Contest



Problems

- A zerorush
- B foodbreak
- C mansion
- D packages
- E javelins
- F audit
- G windowmanager
- H xorpairs
- I runnerups



Copyright © 2025 by The GRCPC 2025 Jury. This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.
<https://creativecommons.org/licenses/by-sa/4.0/>

A zerorush

Time limit: 1s

Zerorush is an exciting single-player game played with a sequence of numbers.

The game begins with a sequence of N integers $a_1, a_2, \dots, a_N$. In each turn, the player chooses any number in the sequence and either increases or decreases it by exactly 2. The game ends successfully when at least one number in the sequence reaches zero.

Your task is to determine the minimum number of moves needed to complete the game.

Input

The first line contains one integer N ($1 \leq N \leq 10^6$) - the length of the sequence.

The second line contains N space separated integers $a_1, a_2, \dots, a_N$ ($-10^9 \leq a_i \leq 10^9$) - the sequence.

Output

One line, the minimum number of moves needed to complete the game, or -1 if it is impossible to do so.

Sample Input 1

6	2
6 5 4 3 4 7	

Sample Output 1**Sample Input 2**

5	-1
3 11 7 13 9	

Sample Output 2

This page is intentionally left blank.

B foodbreak

Time limit: 1s

Alex is organizing the next coding bootcamp at your university. Between managing speakers, coordinating with professors and answering questions in Discord, he's forgotten to plan the most crucial part: lunch breaks.

The bootcamp day starts at time 0 and lasts T units of time. Alex has scheduled N presentations, where presentation i occupies the time interval $[s_i, e_i)$. To arrange the food break, Alex needs to find the largest gap between presentations.

A gap is a maximal time interval $[a, b)$ during the day where no presentations are scheduled (that is, $0 \leq a < b \leq T$ and no presentation overlaps with $[a, b)$). Your task is to find the length of the largest gap.

Input

The first line contains two integers N ($1 \leq N \leq 10^5$) and T ($1 \leq T \leq 10^9$) — the number of presentations and the length of the day.

Each of the next N lines contains two integers s_i and e_i ($0 \leq s_i < e_i \leq T$) — the start and end times of presentation i . Presentation i occupies the half-open interval $[s_i, e_i)$, meaning it includes time s_i but not time e_i .

Output

Output one integer — the length of the largest gap between presentations.

Sample Input 1**Sample Output 1**

4 60 5 10 50 55 25 30 15 20	20
---	----

This page is intentionally left blank.

C mansion

Time limit: 1s

You are the owner of a new mansion and you want to navigate it efficiently.

The mansion has N rooms connected by M one-way security doors. Each door opens only in the specified direction. Every door is either locked or unlocked. Traversing an unlocked door takes 0 time units and traversing a locked door takes 1 time unit. You have a badge that allows you to open any locked door.

Your goal is to travel from room 1 to room N with the minimum total time.

Input

The first line contains two integers N ($2 \leq N \leq 10^5$) and M ($1 \leq M \leq 10^5$) — the number of rooms and the number of doors.

Each of the next M lines contains three integers u_i, v_i ($1 \leq u_i, v_i \leq N, u_i \neq v_i$), and t_i ($t_i \in \{0, 1\}$) — describing a door. The door leads from room u_i to room v_i , and t_i is the traversal time of the door, where 0 means unlocked and 1 means locked.

Output

Output one integer — the minimum total time required to reach room N from room 1, or -1 if room N is not reachable.

Sample Input 1

5 6 1 2 0 2 3 1 3 5 0 1 4 1 4 5 1 2 5 1	1
---	---

Sample Output 1**Sample Input 2**

3 2 1 2 1 2 1 1	-1
-----------------------	----

Sample Output 2

This page is intentionally left blank.

D packages

Time limit: 1s

There are N packages available for delivery, and you want to deliver as many of them as possible given a limited budget.

You can book a single-night courier run, but you must declare exactly how many packages K will be delivered in that run. Since it's peak season, besides the basic cost for each package, the company charges a congestion surcharge. More specifically, each package i has a basic cost a_i and a congestion coefficient b_i . The cost to deliver package i in a run of K packages is $a_i + K \cdot b_i$.

Given a limited budget B , you want to maximize the number of packages delivered.

Input

The first line contains two integers N ($1 \leq N \leq 2 \cdot 10^5$) and B ($0 \leq B \leq 10^9$) - the number of packages and the budget respectively.

The second line contains N space separated integers $a_1, a_2, \dots, a_N$ ($0 \leq a_i \leq 10^5$) - the basic costs for each package.

The third line contains N space separated integers $b_1, b_2, \dots, b_N$ ($0 \leq b_i \leq 10^5$) - the congestion coefficients for each package.

Output

Output two integers - the maximum number of packages that can be delivered within budget and the minimum cost to deliver this number of packages.

Sample Input 1

4 10 1 2 2 1 2 2 2 2	2 10
----------------------------	------

Sample Output 1**Sample Input 2**

5 35 3 2 1 4 5 3 4 5 2 1	3 30
--------------------------------	------

Sample Output 2

This page is intentionally left blank.

E javelins

Time limit: 1s

In a javelin-throwing competition, an athlete faces N targets arranged in a field, numbered from 1 to N . The athlete wants to hit as many targets as possible, but there's a catch: fatigue accumulates with each successful hit.

The athlete has a limited energy budget of B units. Each target i has two properties: a base difficulty a_i and a fatigue coefficient b_i . If target i is the j -th target hit in sequence, the energy cost to hit it is $a_i + (j - 1) \cdot b_i$.

Your task is to determine the maximum number of targets the athlete can hit without exceeding this energy budget. Note that the athlete can choose any subset of targets and hit them in any order, as long as the total cost doesn't exceed E . Every target can be hit at most once.

Input

The first line contains two integers N ($1 \leq N \leq 10^4$) and B ($0 \leq B \leq 10^9$) - the number of targets and the energy budget respectively.

The second line contains N space separated integers $a_1, a_2, \dots, a_N$ ($0 \leq a_i \leq 10^5$) - the base difficulties for each target.

The third line contains N space separated integers $b_1, b_2, \dots, b_N$ ($0 \leq b_i \leq 10^5$) - the fatigue coefficients for each target.

Output

Output two integers: the maximum number of targets the athlete can hit without exceeding the energy budget B and the energy required to do so.

Sample Input 1

5 12 1 1 1 1 1 1 1 1 1 1	4 10
--------------------------------	------

Sample Output 1**Sample Input 2**

6 31 11 6 2 14 3 5 14 1 1 1 4 11	4 25
--	------

Sample Output 2

This page is intentionally left blank.

F audit

Time limit: 1s

You are working as a quality control analyst at a precision manufacturing facility. Your job is to monitor the production line throughout your shift. Today, an internal audit is scheduled from the factory's management.

The production line generates N measurements $a_1, a_2, \dots, a_N$ throughout your shift. The management will issue Q requests. For each request, they give you a range and a number k , and you have to report the minimum, the maximum and the k -th smallest measurement in that range.

Your task is to answer all of their requests. Luckily, you know management well: they only ever choose ranges of two standard lengths, L_1 or L_2 .

Input

The first line contains two integers N ($4 \leq N \leq 10^5$) and Q ($1 \leq Q \leq 10^5$) - the number of measurements and the number of requests respectively.

The second line contains two integers L_1 ($3 \leq L_1 \leq N$) and L_2 ($3 \leq L_2 \leq N$) - the two standard lengths of the ranges.

The third line contains N space separated integers $a_1, a_2, \dots, a_N$ ($0 \leq a_i \leq 10^9$) - the measurements.

Each of the next Q lines contains three integers l ($l \in 1, 2$), i ($1 \leq i \leq N - L_l + 1$) and k ($2 \leq k \leq L_l - 1$) - whether the range length is L_1 or L_2 , the starting index of the range, and the order-statistic the management is interested in, respectively.

Output

Output Q lines. For each request, print three integers: the minimum, the k -th smallest, and the maximum measurement in the specified range.

Sample Input 1

6 5	2 4 9
4 3	2 8 9
9 2 8 4 3 2	2 8 9
1 1 2	2 3 8
1 1 3	2 3 4
2 1 2	
1 3 2	
2 4 2	

Sample Output 1

This page is intentionally left blank.

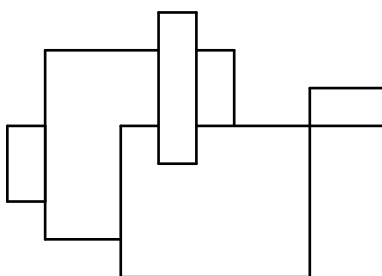
G windowmanager

Time limit: 1s

Bill is developing a window manager for his new operating system.

The scene consists of N rectangles. All windows have distinct integer z coordinates. If windows overlap, the one with greater z coordinates is on top. When drawing the scene, for each window, you only draw the part of it which is on top of all other windows. It turns out that the most time-consuming operation is drawing the borders of the windows.

Bill defines scene complexity as the minimum number of segments that need to be drawn for all the visible window borders. For example, in the following scene:



You need to draw 19 segments: 11 horizontal and 8 vertical ones.

Your task is, given the description of the windows, to calculate how many segments the window manager needs to draw.

Input

The first line of the input contains a single integer N ($1 \leq N \leq 10^5$), the number of windows.

The following N lines describe the windows. Each line contains five integers: x_1, y_1, x_2, y_2, z - coordinates of the bottom-left and the top-right corners, and the z -coordinate of the window. ($0 \leq x_1 < x_2 \leq 1000$, $0 \leq y_1 < y_2 \leq 1000$, $1 \leq z \leq N$, all z are distinct)

Output

One integer, the minimum number of segments that need to be drawn for all visible window borders.

Sample Input 1

```
5
6 4 7 8 4
10 5 12 6 5
3 2 8 7 1
5 1 10 5 2
2 3 3 5 3
```

Sample Output 1

```
19
```

This page is intentionally left blank.

H xorpairs

Time limit: 3s

Alex, a brilliant cryptographer, is on the verge of a major breakthrough. An ancient manuscript speaks of a lost civilization that encoded secrets using a unique numerical system. The key to their cipher lies in two fundamental concepts.

The first is the “**Digital Essence**” of a number, which the manuscript describes as the bitwise XOR sum of all its decimal digits. Let’s call this function $g(x)$. For example, $g(1234) = 1 \oplus 2 \oplus 3 \oplus 4 = 4$, where $\oplus$ denotes the bitwise XOR operation.

The second concept is the “**Mystic Range**” between two numbers, say a and b . This is the sum of the Digital Essence of every integer in the inclusive range between them. We can define this as:

$$f(a,b) = \sum_{i=\min(a,b)}^{\max(a,b)} g(i)$$

Alex has discovered a crucial array of numbers, A , which are the final keys to the cipher. The manuscript’s last instruction is clear: to unlock the ultimate secret, one must calculate the Mystic Range $f(A_i, A_j)$ for every pair of numbers in the array A , and then find the grand total of all these results.

Since the total sum can be very large, you must output the result modulo $10^9 + 7$.

Input

The first line of input contains a single integer N ($2 \leq N \leq 10^5$), the number of elements in the array A .

The second line contains N space-separated integers $A_1, A_2, \dots, A_N$ ($0 \leq A_i \leq 10^9$), the elements of the array.

Output

Print a single integer representing the sum $\sum f(A_i, A_j)$ modulo $10^9 + 7$.

Sample Input 1	Sample Output 1
4 5 15 3 24	356

Sample Input 2	Sample Output 2
5 0 1000000000 0 1000000000 0	758125779

This page is intentionally left blank.

I runnerups

Time limit: 2s

You are working as a sports statistician for a major athletics tournament. Throughout the competition, N athletes have recorded their performance scores $a_1, a_2, \dots, a_N$ in chronological order.

The tournament organizers are interested in analyzing "runner-up excellence" - they want to understand how well the second-best performers did across various event groupings. They will give you Q analysis requests.

For each request, they specify K specific time periods (given as indices $i_1, i_2, \dots, i_K$ in ascending order) that they want to analyze. Your job is to consider every possible subarray that can be formed using these indices as endpoints, find the second-highest score in each subarray, and report the sum of all these runner-up scores.

Input

The first line contains two integers N ($2 \leq N \leq 10^5$) and Q ($1 \leq Q \leq 10^5$) - the number of athletes and the number of analysis requests respectively.

The second line contains N unique space separated integers $a_1, a_2, \dots, a_N$ ($1 \leq a_i \leq 2 \cdot 10^5$) - the performance scores.

Each of the next Q lines starts with an integer K ($2 \leq K \leq N$) followed by K strictly increasing integers $i_1, i_2, \dots, i_K$ ($1 \leq i \leq N$) - the indices of the time periods to analyze. The sum of all K values across all queries is at most $5 \cdot 10^5$.

Output

Output Q lines. For each analysis request, print the sum of the second-highest scores across all possible subarrays formed using the given indices as endpoints.

Sample Input 1

```
9 2
10 12 8 7 4 9 13 15 20
4 2 4 5 9
2 1 9
```

Sample Output 1

```
65
15
```

