

Rain Collection

Input file: **standard input**
 Output file: **standard output**
 Time limit: 5 seconds
 Memory limit: 2048 mebibytes

There are N towers standing side by side in MCO town, the i -th tower (zero-indexed) from the left initially has height H_i . After heavy rain, water might collect on top of the towers. Evirir, the resident dragon of MCO, wonders about how much rainwater these towers can collect.

For a range of towers $[l, r]$ (i.e. towers $l, l + 1, \dots, r$), its *rainfall* is defined as follows:

1. A water column of height $x \geq 0$ can collect on tower j if and only if there exist towers i and k such that $l \leq i \leq j \leq k \leq r$, and both towers i and k are taller than tower j by at least x , i.e.,

$$H_i - H_j \geq x \quad \text{and} \quad H_k - H_j \geq x.$$

2. Define $f(j)$ as the maximum height of a water column that can collect on tower j .
3. The *rainfall* is defined as

$$f(l) + f(l + 1) + \dots + f(r),$$

i.e. the sum of maximum heights of water columns that can collect on these towers.

Evirir has great faith in the new generation of Malaysian OI kids, so asking you to only find the rainfall of one range of towers would be way too easy. Instead, you are to handle Q operations, which can be one of two types:

- Update: $0 \ l \ r \ x$ — add x to all H_i for all i where $l \leq i \leq r$.
- Query: $1 \ l \ r$ — output the rainfall of towers $[l, r]$.

Notes:

- For clarity, when answering a query for the range $[l, r]$, towers outside this range should not be considered when computing $f(i)$ and the rainfall. Towers outside the range cannot be used to trap water.
- The height of a tower can be negative, but they follow the same rules. See the examples for an explanation.

Input

The first line contains two space-separated integers, N and Q .

The second line contains N space-separated integers, H_0, H_1, \dots, H_{N-1} .

Then Q lines follow. Each line represents an operation and contains space-separated integers:

- Update: $0 \ l \ r \ x$ — add x to H_i where $l \leq i \leq r$.
- Query: $1 \ l \ r$ — output the rainfall of towers $[l, r]$.

Output

For every query in order, output the rainfall of the towers in $[l, r]$ on a new line.

Scoring

For all test cases, the input will satisfy the following constraints:

- $1 \leq N \leq 5 \cdot 10^6$
- $1 \leq Q \leq 5 \cdot 10^4$
- $|H_i| \leq 10^7$ for all $0 \leq i \leq N - 1$
- $0 \leq l \leq r \leq N - 1$ for all updates and queries
- $|x| \leq 10^7$ for all updates
- At least one operation is a query.

Subtask	Points	Additional constraints
1	8	$N, Q \leq 1000$
2	8	$Q = 1$
3	16	$N \leq 10^6$ and there are no updates in the input
4	18	$l = r$ and $x > 0$ for updates. $[l, r] = [0, N - 1]$ for queries.
5	25	$N \leq 5 \cdot 10^5$
6	25	—

Examples

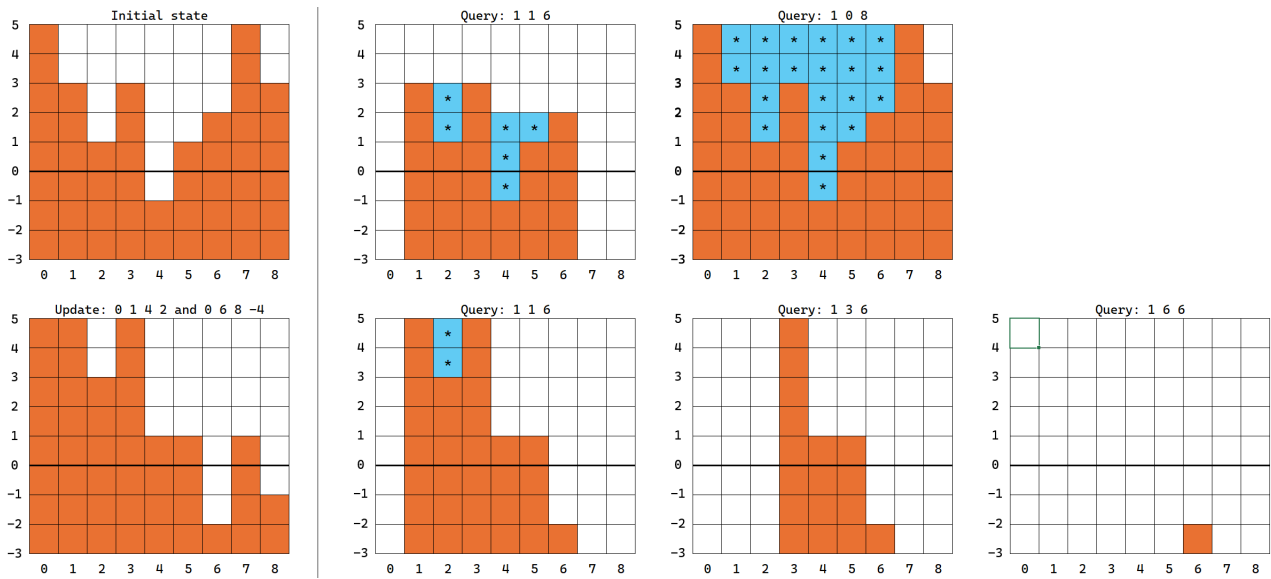
example input	example output
<pre> 9 7 5 3 1 3 -1 1 2 5 3 1 1 6 1 0 8 0 1 4 2 0 6 8 -4 1 1 6 1 3 6 1 6 6 </pre>	<pre> 6 21 2 0 0 </pre>
<pre> 5 6 -2 3 1 4 2 0 0 2 1 0 0 4 3 0 3 4 8 0 0 0 10 0 1 3 1 1 0 4 </pre>	<pre> 10 </pre>

Note

Example 1

This example is valid for subtasks 1, 5, and 6.

There are $N = 9$ towers. Here is a visualisation of the updates and queries:



A * denotes part of a water column.

In the first query 1 1 6, towers 1 to 6 are considered. Consider tower $j = 5$ which has height 1. A water column of height 1 can collect on tower 5 because:

- Tower $i = 3$ has height 3, which is taller than tower 5 by 2.
- Tower $k = 6$ has height 2, which is taller than tower 5 by 1.

There cannot be a water column of height 2 on tower 5 because there is no tower k where $j \leq k \leq 6$ that is taller than tower 5 by at least 2 (i.e. has height at least $1 + 2 = 3$). Note that you cannot take $k = 7$ because k is not in the query's range $[1, 6]$. Therefore, $f(5) = 1$, represented by the 1 water cell on tower 5.

In the second query 1 0 8, towers 0 to 8 are considered. Consider tower $j = 4$ which has height -1 . A water column of height 6 can collect on tower 4 because towers $i = 0$ and $k = 7$ have height 5, which are taller than tower 4 by 6. It can also be proven that 6 is the maximum possible height, so $f(4) = 6$.

In the update 0 1 4 2, the height of towers 1 to 4 are increased by 2. In the update 0 6 8 -4, the height of towers 6 to 8 are decreased by 4.

In the query 1 6 6, note that a tower with negative height still needs taller towers around it to trap water.

Note that there can always be a water column of at least height 0 on a tower by choosing $i = j = k$.

Example 2

This example is valid for subtasks 1, 5, and 6.