Databases

Pepe has n **databases** numbered from 1 to n. He also has $2 \cdot 10^5$ **data chunks** numbered from 1 to $2 \cdot 10^5$. The ith database can store at most $c[i]$ data chunks. Initially all the databases are empty.

Pepe's weird algorithm will do q operations on the databases. In the ith operation, from all the databases numbered from $l[i]$ to $r[i]$, the algorithm will add data chunk $x[i]$ to the end of the database. But if after this operation, any database exceeds its capacity then it will delete the data chunk at the start of that database (i.e. databases have a queue like structure).

After all the q operations, Pepe realizes that some databases contain the same data chunk multiple times. But Pepe is only interested in distinct data chunks. So for each database i, print how many distinct data chuncks it contains in the end.

Input

The input will be given in the following format:

- line 1: n q
- line 2: $c[1]$ $c[2]$ $\cdots$ $c[n]$
- line $2+i$ $(1 \leq i \leq q)$: $l[i]$ $r[i]$ $x[i]$

Output

Print the output in the following format:

- line i $(1 \leq i \leq n)$: number of distinct data chunks ith database contains in the end

Constraints

- $1 \leq n, q \leq 2 \cdot 10^5$
- $1 \leq c[i] \leq 2 \cdot 10^5$ for $(1 \leq i \leq n)$
- $1 \leq l[i] \leq r[i] \leq n$ for $(1 \leq i \leq q)$
- $1 \leq x[i] \leq 2 \cdot 10^5$ for $(1 \leq i \leq q)$

Subtasks

1. (10 points) $n, q \leq 2000$
2. (11 points) all $x[i]$ are distinct
3. (15 points) $c[i] = 2 \cdot 10^5$ for $(1 \leq i \leq n)$

4. (5 points) $x[i] \leq 2$ for $(1 \leq i \leq q)$
5. (7 points) $x[i] \leq 50$ for $(1 \leq i \leq q)$
6. (12 points) $c[i] \leq 50$ for $(1 \leq i \leq n)$
7. (17 points) $c[i] = c[j]$ for $(1 \leq i, j \leq n)$
8. (23 points) No additional constraints.

Examples

Example 1

```
3 4
1 2 3
1 2 3
1 2 1
2 3 1
3 3 2
```

The correct output is:

```
1
1
2
```

Here is what happened in every operation (green means the data chunk is added, red means the data chunk is deleted):

- at start: $[]$, $[]$, $[]$
- operation 1: $[3]$, $[3]$, $[]$
- operation 2: $[3, 1]$, $[3, 1]$, $[]$
- operation 3: $[1]$, $[3, 1, 1]$, $[1]$
- operation 4: $[1]$, $[1, 1]$, $[1, 2]$

In the end, the databases contain 1, 1 and 2 distinct data chunks respectively.