Problem K. No more regrets

Time limit:	4 seconds
Memory limit:	1024 megabytes

After the provincial team selection, White fell into a long period of disappointment. She could not find any hope in her results. Even so, White continued her final training before the NOI. Besides her regular training, she also began to explore topics she had never had the chance to learn during her OI days — hoping that, before saying goodbye, there would be no more regrets.

Shaking off her wandering thoughts, White suddenly focused on a problem in front of her, a plain and boring data structure problem —

White has a sequence of n integers $a_1, a_2, \ldots, a_n$. She will perform q operations on this sequence. Each operation is one of the following three types:

- `1 l r v` — Add v to each element in the interval $[l, r]$.
- `2 l r v` — Assign each element in the interval $[l, r]$ to v.
- `3 l r` — Query the value $\sum_{i=l}^{r}(\min_{j=l}^{i} a_j) \times (\max_{j=l}^{i} a_j)$ modulo 2^{64}.

Your task is to simulate all operations and output the results of all queries.

Input

The first line of the input contains two integers n, q ($1 \le n, q \le 2 \times 10^5$), representing the number of elements and the number of operations.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($0 \le a_i \le 10^9$), representing the initial elements.

Each of the next q lines describes an operation, in one of the three formats:

- `1 l r v` ($1 \le l \le r \le n, -10^9 \le v \le 10^9$), representing the Add operation.
- `2 l r v` ($1 \le l \le r \le n, 1 \le v \le 10^9$), representing the Assign operation.
- `3 l r` ($1 \le l \le r \le n$), representing the Query operation.

It's guaranteed that $0 \le a_i \le 10^9$ for each $1 \le i \le n$ during the whole process.

Output

For each Query operation, print an integer in a single line—the result of the query modulo 2^{64}.

Examples

standard input	standard output
5 8 2 3 5 4 1 3 1 5 3 2 4 2 4 5 2 3 1 5 3 2 4 1 1 2 5 3 1 5 3 2 4	35 39 40 34 177 120
10 20 1 2 3 4 5 6 7 8 9 10 1 1 10 1 3 1 5 3 2 9 3 8 10 1 2 5 10 3 1 5 3 2 9 3 8 10 1 5 9 -5 3 1 5 3 2 9 3 8 10 1 2 5 -10 3 1 5 3 2 9 3 8 10 1 5 9 5 3 1 5 3 2 9 3 8 10	40 156 270 120 1202 270 118 831 80 33 61 80 40 156 270

Note

In the first testcase:

The original sequence is `2 3 5 4 1`. After the third operation, it becomes `2 3 2 2 1`, and after the sixth operation, it becomes `7 8 7 7 6`.

For the first query, the answer is $\sum_{i=1}^{5}(\min_{j=1}^{i} a_j) \times (\max_{j=1}^{i} a_j) = 2\times2+2\times3+2\times5+2\times5+1\times5 = 35$.

For the second query, the answer is $\sum_{i=2}^{4}(\min_{j=2}^{i} a_j) \times (\max_{j=2}^{i} a_j) = 3\times3+3\times5+3\times5 = 39$.