Inaho II

Inaho is a scientist. He discovered the fourth dimension recently. In fact, he also discovered the fifth, the sixth, the seventh, the eighth, the ninth, and the tenth dimension! Today, as he was travelling on the infinite plane of uniform density, he fell into an N-dimensional hole. "How did that even happen?" he asks himself in disbelief, but quickly realizing that finding his way out is perhaps more important. However, as a measly 3-dimensional being, he cannot understand the complexities of N dimensions.

Fortunately, he knows the way out is the direction with the least density. However, the densities are constantly changing, so he needs to react quickly. Inaho does not know the initial densities, but in a desperate attempt to escape, he assumes them to be 0.

He will tell you Q statements, which will either ask you for the sum of the density in a range, or tell you that the density of a certain position changes to x. Please help him!

Input Specification

The first line will contain two space-separated integers N $(1 \le N \le 10)$, and Q $(1 \le Q \le 10^5)$, the number of dimensions and the number of statements respectively.

The second line will contain N space-separated integers that represent the size of each dimension, $(l_1, l_2, \ldots, l_N)$ $(1 \le l_1 \times l_2 \times \ldots \times l_N \le 10^7)$.

The next Q lines will contain one of the following statements:

- $1\ a_1\ a_2\ \ldots\ a_N\ x$, which represents a change at position a $(1 \le a_i \le l_i)$ to x $(0 \le x \le 2^8 - 1)$.
- $2\ a_1\ a_2\ \ldots\ a_N\ b_1\ b_2\ \ldots\ b_N$, which represents summing the total density between a $(1 \le a_i \le l_i)$ and b $(a_i \le b_i \le l_i)$ inclusive.

Subtasks

For 2 of the 25 available marks, $N \le 3$.

For an additional 2 of the 25 available marks, N = 4.

For an additional 2 of the 25 available marks, $Q \le 100$.

For an additional 2 of the 25 available marks, $l_1 \times l_2 \times \ldots \times l_N \le 1\,000$.

NOTE: This problem is practically *impossible* to solve with Python (even with PyPy). To allow a correct Python solution to pass would require a time limit of 60 seconds, which is unrealistic. It is recommended to use C++.

Output Specification

For each summation statement, output the answer on a separate line.

Sample Input 1

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

Sample Output 1

```
2
15
18
```

Sample Input 2

```
10 9
3 3 3 3 3 3 3 3 3 5
1 1 1 1 1 1 1 1 1 2 1 13
1 3 3 3 3 3 3 3 3 3 5 19
1 1 1 3 2 2 2 2 2 2 2 3
2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 1 24
2 1 1 1 1 1 1 1 1 2 1 3 3 3 3 3 3 3 3 3 5
2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 5
1 1 1 1 1 1 1 1 1 2 1 12
2 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3
```

Sample Output 2

```
13
35
19
39
```