### Overflow

#### **Time limit:** 1.4s **Memory limit:** 256M

Joe is an economical man and he has found a new scheme to save some money! Joe wants to save money on his water bill, so he decides to collect rainwater. Joe has a row of N water containers, all initially empty and each with a maximum capacity of  $v_i$  liters of water. The containers are connected in such a way that when container i overflows, the excess liquid flows to container i+1. When container N overflows, the extra water magically disappears. Joe wishes to gather information about the efficacy of his setup, and so he has prepared Q queries of the following types:

- 1 l r v: Rainfall arrives, and v liters of water falls directly into each container from position l to r inclusive.
- 2 x v: Joe changes the maximum capacity of bucket x to v. If the volume decreases, any resulting overflow passes on to bucket x+1 as usual.
- 3 x : Joe wishes to know the current volume of water being held in container x.

#### **Constraints**

 $1 \leq N,Q \leq 10^5$ 

 $1 \leq x_i \leq N$ 

 $1 \leq l_i \leq r_i \leq N$ 

 $1 \le v_i \le 10^{12}$ 

#### **Subtask 1 [10%]**

 $1 \leq N, Q \leq 100$ 

### **Subtask 2 [10%]**

All queries of type 3 appear after all queries of type 1; there are no queries of type 2.

#### **Subtask 3 [30%]**

For all queries of type 1,  $l_i=r_i$ .

#### **Subtask 4 [50%]**

No additional constraints.

### **Input Specification**

The first line will contain N and Q, the number of containers and number of queries respectively.

The second line will contain N space-separated integers,  $v_1, v_2, \ldots, v_N$ .

The next Q lines will contain queries of the form mentioned in the problem description.

# **Output Specification**

Output the answer to each type 3 query on separate lines.

# **Sample Input**

```
5 5
4 5 2 9 3
1 1 2 3
2 1 1
1 2 5 2
3 4
3 5
```

# **Sample Output**

```
4
2
```