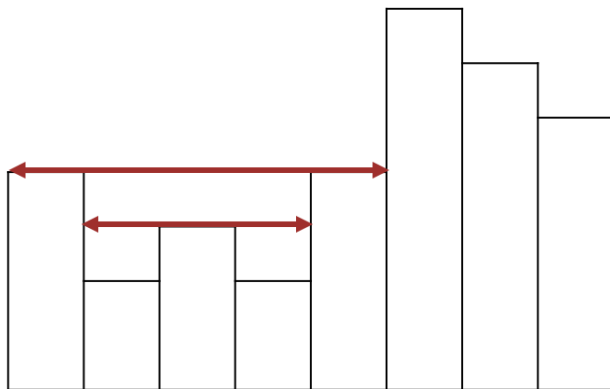


# Skyscraper

## Problem ID: skyscraper

There are  $n$  skyscrapers. They are arranged in a row from left to right and numbered from 1 to  $n$ . Skyscraper  $i$  has height  $h_i$ . We say that skyscraper  $i$  can see skyscraper  $j$  if there is no skyscraper higher than skyscraper  $i$  between the two skyscrapers and  $h_j \leq h_i$ .

For example, assuming that  $n = 8$  and  $h = [4, 2, 3, 2, 4, 7, 6, 5]$ . For the skyscraper 3, it can see 3 skyscrapers from skyscraper 2 to skyscraper 4. For the skyscraper 5, it can see 5 skyscrapers from skyscraper 1 to skyscraper 5. This is represented in the figure below.



There's a lot of construction, and we'd like to keep track of which skyscrapers can see each other. In particular, we'd like to build a program supporting three types of queries:

- `num_visible_skyscrapers(i)`: returns the number of skyscrapers that skyscraper  $i$  can see.
- `set_height(i, x)`: updates the height of skyscraper  $i$  to  $x$ . (i.e.  $h_i = x$ )
- `set_height_range(i, j, x)`: updates the heights of skyscrapers from  $i$  to  $j$  to  $x$ . (i.e.  $h_i = h_{i+1} = \dots = h_j = x$ ) where  $i \leq j$ .

You are given initial heights the skyscrapers and  $q$  queries. These  $q$  queries should be processed in an order. Write a program to handle these queries, and print any result from `num_visible_skyscrapers`.

## Input

Your program will receive input from standard input.

The first line contains two space-separated positive integers  $n$  and  $q$  representing the number of skyscrapers and the number of queries. The next line contains  $n$  positive integers. The  $i$ -th integer represents the initial height of skyscraper  $i$ ,  $h_i$ . In the following  $q$  lines, the  $i$ -th line contains several positive integers representing the  $i$ -th query. Each line is in one of the three following formats:

- 1 i: represents `num_visible_skyscrapers(i)`
- 2 i x: represents `set_height(i, x)`
- 3 i j x: represents `set_height_range(i, j, x)`

You may assume that there is at least one `num_visible_skyscrapers` query.

## Output

Your program should write to standard output.

Print  $b$  lines, one line for each of the  $b$  `num_visible_skyscrapers` calls, each containing the result of that.

## Constraints

- $1 \leq n, q \leq 3 \cdot 10^5$
- $1 \leq h_i \leq 10^9$
- $1 \leq i \leq n$  for num\_visible\_skyscrapers and set\_height queries
- $1 \leq i \leq j \leq n; 1 \leq x \leq 10^9$  for set\_height\_range queries

## Subtasks

You will get points for each subtask when you pass all of the testcases of the subtask.

1.  $n, q \leq 2 \cdot 10^3$  (11 points)
2. Only num\_visible\_skyscrapers is given (23 points)
3. Only num\_visible\_skyscrapers and set\_height are given (37 points)
4. No additional constraints (29 points)

### Sample Input 1

```
8 6
4 2 3 2 4 7 6 5
1 3
1 2
2 3 8
1 5
3 5 7 1
1 8
```

### Sample Output 1

```
3
1
2
5
```