

Borrow Classrooms

Problem Description

During college, it is often necessary to rent classrooms. It is necessary to ask the university to borrow classrooms for everything from department events to study groups for self-study discussions. The size and function of the classrooms are different, and the identity of the person borrowing the classroom is different, thus the procedure for borrowing the classroom is also different.

Faced with the huge amount of information about renting classrooms, we naturally want to program to solve this problem.

We need to process the information on classroom booking for the next n days, where on day i the school has r_i classrooms available to rent. There are m orders in total, and each order is described by three positive integers d_j, s_j, t_j , which means that some renter needs to rent classrooms from day s_j to day t_j (including day s_j and day t_j), and d_j classrooms need to be rented every day.

We assume that the renter has no requirement on the size or location of the classroom. That is, for each order, we only need to provide d_j classrooms per day, regardless of which classrooms they are and whether they are the same classrooms every day.

The principle of borrowing classrooms is first come first served, which means we need to assign classrooms to each order in the order of the order. If we encounter an order that cannot be fulfilled during the allocation process, we need to stop the allocation of classrooms and notify the current applicant to modify the order. Unsatisfied here means that at least one day from day s_j to day t_j has less than d_j classrooms left.

Now we need to know if there will be orders that cannot be fulfilled. If so, which application needs to be notified to modify the order?

Input

The first line contains two positive integers n and m , representing the number of days and the number of orders.

The second line contains n positive integers, where the i_{th} number is r_i , which represents the number of classrooms available for rental on the i_{th} day.

Then there are m lines, each containing three positive integers d_j, s_j , and t_j , representing the number of rooms rented, the starting day, and the ending day.

Every two adjacent numbers are separated by a space. Days and orders are numbered with integers starting from 1.

Output

If all orders could be satisfied, the output has only one line and contains an integer 0. Otherwise (the order cannot be fully satisfied) output two lines. In the first line, output the negative integer -1. And in the second line, output the serial number of the applicant whose order needs to be modified.

Sample Input

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

Sample Output

```
-1
2
```

Hint

[Explanation of Sample]

After the first order is satisfied, the numbers of classrooms remaining in 4 days are 0, 3, 2, and 3, respectively. The second order requested 3 classrooms per day from day 2 to day 4, and the number of classrooms remaining on day 3 was 2, so this order could not be satisfied. The allocation stops and the 2nd applicant is notified to revise the order.

[Data range]

For 10% of the data, $1 \leq n, m \leq 10$;

For 30% of the data, $1 \leq n, m \leq 1000$;

For 70% of the data, $1 \leq n, m \leq 10^5$;

For 100% of the data, $1 \leq n, m \leq 10^6, 0 \leq r_i, d_j \leq 10^9, 1 \leq s_j \leq t_j \leq n$.