Mock INOI #7

Statement is not available on English language

B. Lazy Setter

3 seconds, 256 megabytes

Just as the No Names begin to take a foothold once again, the entire East Side is shaken by another attack. This time, they are prepared and are holding out for the time being, but the issue is that humans lack in stamina to mount a sustained defence. As a result, it is decided that the defences will be automated. Building the machines itself is a trivial task, as there are a great number of powerful magic users who can easily do the job. However, there is a need to program the machines so that they can run the defence without human intervention. This is far more difficult as there is a dearth of programming experience in the No Name community. However, it can simultaneously be considered the most crucial aspect, as if the programmer makes a mistake, it can easily lead to the machines turning on the creators - a worse outcome than if they had stuck to manual defence from the beginning. Hence, a decision is made to find the best algorithm experts and programmers among the No Names (and neighbouring communities) through a contest: the first [No Name] Internal Olympiad in Informatics (IOI). Of course, Sakamaki Izayoi (being the most intelligent) is the sole author of the problems in this contest, but as he is also the linchpin of the ongoing manual defence, he cannot afford to spend too much time on writing the contest. After writing 5 out of 6 problems, he is running out of time, and to save his time he decides that the last problem shall have **random testdata**. The problem is below. Izayoi can easily solve it, but can you?

There is a sequence a of length n: $a_1, a_2, \ldots, a_n$. m actions are carried out on this sequence. Each action is of one of two types - update or query. Each update is defined by three integers l,r,v, meaning that a_i will be set to v for all $l \le i \le r$. Each query is defined by two integers l,r, meaning that you want to find the number of *inversions* in the range $[l, r]$. In other words, you want to find the number of pairs (i, j) such that $l \le i < j \le r$ and $a_i > a_j$.

As mentioned before, this problem has random testdata. More specifically, each value in the input is randomly selected from the space of possible values. For example, all a_i are selected uniformly from the range $[1, 10^9]$, and unless specified otherwise updates and queries occur with equal probability.

Input

On the first line, there are three integers - ST, n and m. ST is the subtask number of the testcase. On the second line, there are n integers - $a_1, a_2, \ldots, a_n$.

Each of the next m lines follows one of the following formats:

- $1, l, r, v$ denoting that we set all values in the range $[l, r]$ to v.
- $2, l, r$ denoting a query for the number of inversions in $[l, r]$.

Output

On the i-th line, output a single integer: the answer for the i-th query.

Scoring

For all subtasks, $1 \le n, m \le 2 \times 10^5$, $1 \le a_i, v \le 10^9$, $1 \le l \le r \le n$.

- Subtask 1 [9 points]: $n = 1$
- Subtask 2 [15 points]: $n, m \le 100$
- Subtask 3 [14 points]: $n, m \le 7000$
- Subtask 4 [6 points]: There are no queries.
- Subtask 5 [12 points]: There are no updates, $n \le 5000$, $m \le 3 \times 10^4$
- Subtask 6 [8 points]: There are no updates, $n, m \le 3 \times 10^4$
- Subtask 7 [36 points]: No additional constraints.

input

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

output

```
3
1
0
4
```

C. Peaceful Times

6 seconds, 1024 megabytes

Peace has returned to Little Garden, and the No Names decide to play ball games to pass the time. Unfortunately, after the game ends, the balls used have to be cleaned up and put back into the boxes whence they came from.

The No Names have n different colours of balls, and for each colour i there are b_i balls. It is guaranteed that there is at least one ball of each colour. In addition, they have m boxes, and each can fit k balls. It is guaranteed that $\sum b_i = m \times k$ - that is, the total number of balls is equal to the total capacity of the boxes.

The No Names want the balls to have some sort of arrangement instead of them being haphazardly strewn everywhere. Therefore, the goal of the No Names is to place the balls into boxes such that in each box there are at most 2 colours of balls. Can you help them do this?

Note: For some unknown reason (one of the many mysteries of Little Garden), it seems that $m \ge n - 2$ - that is, the number of colours is at most 2 more than the number of boxes.

Input

There are multiple test cases for each input. The first line of the input contains T, the number of test cases.

For each test case, the first line contains n, m, and k - the number of colours, the number of boxes, and the capacity of each box. The second line contains n integers - the i-th integer represents the number of balls of colour i, b_i.

Output

For each test case:

- If there is no valid solution, output -1 on a single line.
- Otherwise, output m lines. Each line consists of 4 integers a, b, c, d denoting that you are placing b balls of colour a and d balls of colour c in the i-th box. Note that if you wish to place only one colour of ball in the box, simply let b or d be 0.

Scoring

For all subtasks, $1 \le T \le 10$, $1 \le n \le 500$, $n - 2 \le m \le 5000$, $m \ge 1$, $1 \le k \le 5000$, $\sum b_i = m \times k$

- Subtask 1 [5 points]: $m = 1$
- Subtask 2 [14 points]: $m = 2$, $k = 5$
- Subtask 3 [5 points]: $m = 3$, $k = 5$
- Subtask 4 [11 points]: $k = 2$
- Subtask 5 [24 points]: $m = n - 1$
- Subtask 6 [8 points]: $m \ge n - 1$

- Subtask 7 [17 points]: $n \leq 50$, $k \leq 500$
- Subtask 8 [14 points]: $n \leq 100$
- Subtask 9 [2 points]: No additional constraints.

input

```
4
1 1 10
10
4 3 100
80 30 90 100
5 3 1000
200 400 500 900 1000
6 4 100
25 30 50 80 95 120
```

output

```
1 10 0 0
1 80 2 20
2 10 3 90
4 100 0 0
-1
1 5 5 95
1 20 4 80
2 30 6 70
3 50 6 50
```