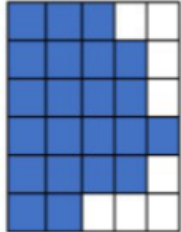


13205 Antivirus

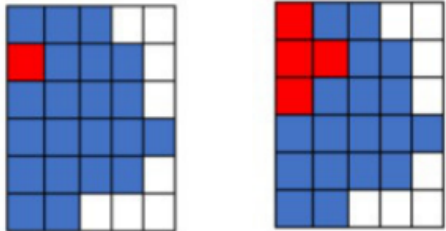
Machine Learning techniques applies to a many fields quite know, including the detection of computer viruses. In this occasion we will analyze the data produced by the C.C.C (Company of Capital Consumption).

The files of the C.C.C. are disposed in contiguous positions. In each position there is a set of cells with records. We can imagine these registers as a list of lists (List of files in this case).



It is well know that exists a virus that will infect the system, and this virus in particular has a very special way of infect archives. Using some heuristics has been detected that exists two versions of the virus (type A and type B). The virus starts to infect files from left to right, in other words, always starts with the register in the leftmost position and keeps moving to the right.

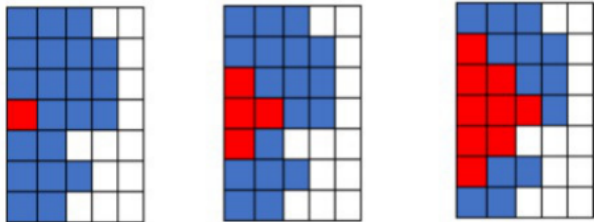
The virus type A selects an initial position i and starts to consume registers from that position and forms a kind of triangle (always form the largest possible triangle). We can show the virus' growth with the next graph: Be the starting position = 2 (The red cells are the infected registers).



Day 1 and day 2

The virus type A won't infect more registers because it cannot growth more (there are no more registers up, to extend the triangle).

Another example: Be the starting position = 4
The virus type A will not infect more registers because there are no more free registers to the right.



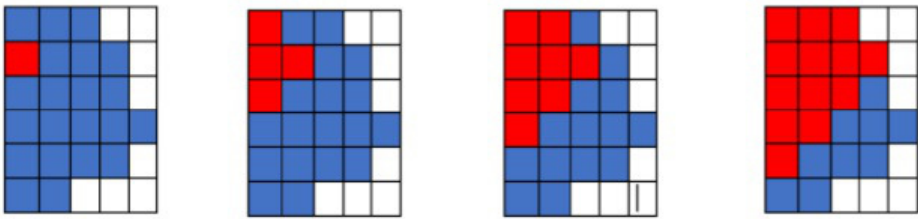
Day 1, day 2 and day 3

In summary the virus type A tries to form a triangle centered in the initial position i , the triangle must be complete (no missing parts). Every file j located upper of position i must have an infected record less than the file in the position $j + 1$.

Analogously, each file j located down of the position i must have an infected record less than the file in the position $j - 1$.

The virus type B acts in a similar way, it selects an initial position i and starts to consume registers from that position and forms a kind of incomplete triangle (always forms the largest possible incomplete triangle).

We can show the growth of the virus type B with the next graph: Be the initial position = 2 (red cells are infected records)

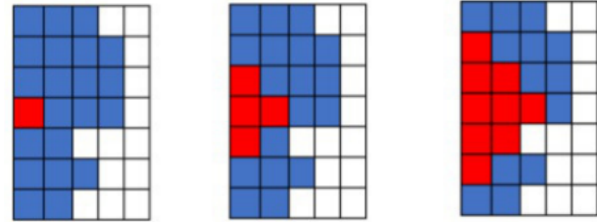


Day 1, day 2, day 3 and day 4

The virus type B will not infect more registers because there are no more free registers to the right.

Another example: Be the initial position = 4

The virus type B will not infect more registers because there are no more free registers to the right. In summary, the virus type B tries to form an incomplete triangle centered in the initial position i . It is understood that the incomplete parts refer to the upper and bottom of the triangle. The corner that grows from left to right should always be complete. Every file j located upper of position i must have an infected record less than the file in the position $j + 1$.



Day 1, day 2 and day 3

Analogously, each file j located down of the position i must have an infected record less than the file in the position $j-1$.

Knowing the way that the virus expands, we want to know how many records won't be infected in a certain range (we mean the range of infected files).

Input

The input may contain several test cases.

Each input case begins with a line that contains N ($1 \leq N \leq 5 \cdot 10^5$), the number of files. The next line contains N integers r_i ($1 \leq i \leq N$ and $1 \leq r_i \leq 10^9$), the amount of registers of each file.

Next comes an integer Q ($1 \leq Q \leq 5 \cdot 10^5$), the number of queries. After that, Q lines follows, specifying the queries in the following format:

- 1 i val = Replace the element in the position i ($1 \leq i \leq N$) with the value val ($1 \leq val \leq 10^9$)
- 2 i $type$ = of the total of registers of the infected files, calculate how many registers won't be infected if the virus starts to infect the file i . A file is consider infected if it has at least one infected record.

For every query of type 2 you must assume that the system is not infected. The variable $type$ indicates the virus type. If $type = 1$ the virus is of type A, in other case the virus is of type B.

In every case you must assume that the propagation will last at least r_i days (the virus will infect the maximum possible quantity of files).

Output

For every query of type 2 print a single line with two integers: the maximum quantity of infected registers (of the infected files), and the quantity of registers that are not infected. See the example input and output for more detail.

Explanation:

In the first query, the virus type is B, therefore it extends until infect all the records of file 6. Files 5, 6, and 7 are infected. One record of the file 5, two records of the file 6 and one record of the file 7 are infected. Thus $(6-1) + (2-2) + (5-1) = 9$ uninfected records remain.

In the second query, the virus type of type A grows in the same manner than the first query.

In the third query, the virus of type B expands until consumes 4 records of the 4 file. The virus no longer progresses because there are no records to infect in the 6 file (its two registries have already been infected).

In the fourth query the virus of type A grows in the same manner than the third query.

In the fifth query we change the element in the position 6 with the value 6.

In the sixth query, files $[1 \dots 11]$ get infected. The amount of infected records are: 1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1. The maximum is 6. 14 records remains uninfected.

In the seventh query, the virus A grows in the same manner than the last query.

In the eighth query, the virus type B consumes all the records of the file 4. The files [1...8] get infected and the amount of infected records are: 2, 3, 4, 5, 4, 3, 2, 1. The maximum is 5. 14 records remains uninfected.

In the ninth query the virus type A infect 4 records of the 4 file. After that it can no longer expands. The maximum is 4. Files [1...7] are infected. 18 records remains uninfected.

Sample Input

```
12
3 4 5 5 6 2 5 4 3 2 1 1
9
2 6 0
2 6 1
2 4 0
2 4 1
1 6 6
2 6 0
2 6 1
2 4 0
2 4 1
```

Sample Output

```
2 9
2 9
4 14
4 14
6 8
6 8
5 14
4 18
```