In Flatland, the river Big Flat, rich in fish, flows. Many years ago, the river was divided between n fishing enterprises, each of which received a continuous section of the river. At the same time, the i-th enterprise, if we consider them in order, starting from the source, initially received a section of the river with length a_i.

Since then, various events have occurred with fishing enterprises in Flatland k times. Each of the events was one of two types: the bankruptcy of a certain enterprise or the division of a certain enterprise into two.

In some events, a section of a river belonging to the enterprise with which this event occurs is divided into two parts. Each such segment has a length greater than or equal to 2. Division occurs according to the following rule. If the segment has an even length, then it is divided into two equal parts. Otherwise, it is divided into two parts, the lengths of which differ exactly by one, while the part that is closer to the source of the river has a shorter length.

With bankruptcy of an enterprise, the following occurs. The section of the river that belonged to the bankrupt enterprise is transferred to its neighbors. If the bankrupt enterprise has one neighbor, then the entire section of the river of the bankrupt enterprise is transferred to this neighbor. If there are two neighbors, then the segment of the river is divided into two parts as described above, after which each of the neighbors joins the closest part to its segment.

When dividing an enterprise, the river segment that belonged to the dividing enterprise is always divided into two parts as described above. The divided enterprise is liquidated and two new enterprises are formed.

Thus, after each event, each enterprise owns a certain section of the river.

The Ministry of Finance of Flatland proposes to introduce a tax on fishing enterprises proportional to the square of the length of the river segment belonging to the respective enterprise. In order to analyze how this tax will work, the minister wants to know from the available data how the value equal to the sum of the squares of the lengths of river sections belonging to enterprises changed after each event.

It is required to write a program that, according to a given initial separation of the river between enterprises and a list of events that occurred with enterprises, will determine what is the sum of the squares of the lengths of river segments belonging to enterprises at the initial time and after each event.

The first line of the input contains a single integer n – the initial number of enterprises ($2 \le n \le 100$). The second line of the input file contains n integers $a_1, a_2, \ldots, a_n$ – lengths of the initial sections of the river. The third line of the input file contains an integer k – the number of events that occurred with the enterprises ($1 \le k \le 100000$).

The subsequent k lines contain descriptions of events, the i-th line contains two integers: e_i and v_i – the type of event and the number of the enterprise with which it occurred. Value $e_i = 1$ means that the enterprise, which after all previous events is v_i-th in order, if you count from unity from the source of the river, it went bankrupt, and the value $e_i = 2$ means that this company is divided into two.

It is guaranteed that for all i from 1 to $k - 1$ the condition is satisfied: $|v_i - v_{i+1}| \le 200$.

It is guaranteed that the value of v_i does not exceed the current number of enterprises. **It is guaranteed** that if a section of an enterprise during bankruptcy or division is required to be divided into two parts, then it has a length greater than or equal to 2. **It is guaranteed** that if a single enterprise remains on the river, it will not go bankrupt.

The output file must contain $(k + 1)$ integers, one per line. The first line should contain the initial sum of the squares of the lengths of the river segments, and each of the next k lines should contain the sum of the squares of the lengths of the river segments after the next event.

Sample input:

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

Sample output:

```
75
105
73
101
```

83
113