H Horrendous Mistake

Time limit: 4s

While looking through the Grand Archive of Problematic Code, you found this horrendous mistake in one of the code snippets. In this code, they tried to calculate the sum of an array, but got indices and values mixed up. The code of this `sum` function is shown in Figure H.1.

```cpp
long long sum(vector<int> a) {
    long long ans = 0;
    for (int x : a)
        ans += a[x];
    return ans;
}
```

```python
def sum(a: list[int]) -> int:
    ans = 0
    for x in a:
        ans += a[x]
    return ans
```

```java
long sum(int[] a) {
    long ans = 0;
    for (int x : a)
        ans += a[x];
    return ans;
}
```

```kotlin
fun sum(a: List<Int>): Long {
    var ans = 0L
    for (x in a)
        ans += a[x]
    return ans
}
```

Figure H.1: The function `sum` intends to calculate the sum of an array, but `x` refers to a value in the array, instead of an index. The code is shown in C++ and Python in the top row, and Java and Kotlin in the bottom row.

You wonder how this function behaves exactly, and decide to thoroughly test it. Starting with some initial array, you apply a sequence of updates. For each update, you change one of the values of the array. You wonder what the value of the function is after each update.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 2 \cdot 10^5$), the length of the array.
- One line with n integers a ($0 \leq a < n$), the values in the initial array.
- One line with an integer t ($1 \leq t \leq 2 \cdot 10^5$), the number of updates in your sequence of testing.
- t lines, each with two integers x and v ($0 \leq x, v < n$), indicating that the xth entry in the array is updated to the new value v.

Note that the array uses 0-based indexing.

Output

For each update, output the return value of the function `sum` applied to the array after applying the update.

Sample Input 1

```
5
0 0 3 2 0
4
1 1
0 4
3 4
1 4
```

Sample Output 1

```
6
10
9
8
```

Sample Input 2

```
5
4 2 2 4 2
3
0 1
4 3
2 3
```

Sample Output 2

```
10
13
16
```