Neetu and 2G

Neetu, a professional detective and a high functioning sociopath, gets a lot of cases by Inspector Lestrade and obviously, he doesn't have the time to solve all of them. Neetu, in most cases, just keeps wondering how much time it'll take him to solve the cases. 2G is worried by it and he sometimes helps Neetu by solving his **most recent** case.
Initially Neetu has no cases, but Lestrade can give him a case at any time represented by "1 x" where "x" is the number of millisecond it'll take Neetu to solve the case.
Sometimes when 2G decides to help by solving the most recent case, it is represented as "2"
Every time Lestrade gives Neetu a case, or 2G solves one, you have to output 2 space separated integers in a new line, the minimum time it'll take Neetu to solve any unsolved case, and total time for Neetu to solve all unsolved cases. Your task is to help Neetu calculate these values. In case there is no case to solve, minimum time to solve a case is -1.

Input:

First line contains T – number of test cases.
Every testcase has Q – the number of queries (Either Lestrade giving a new case, or 2G solving one) Q lines follow. Every line is from:

- 1 x – Lestrade gives Neetu a case that'll take him x milliseconds to solve.
- 2 – 2G solves Neetu's most recent case, if he has one.

Output:

Every testcase has exactly Q lines of output.
Each line has 2 space separated integers. Minimum time for Neetu to solve any unsolved case, Total time for Neetu to solve all unsolved cases.

Constraints:

$1 <= T <= 1,00,000$
$1 <= Q <= 1,00,000$
$1 <= X <= 10^9$
Sum of Q across all test cases <= 1,00,000

Sample Input:

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

Sample Output:

```
5 5
3 8
3 12
3 8
2 10
3 8
5 5
```