

Problem H. Hash Table

Input file: *standard input*
Output file: *standard output*
Time limit: 5 seconds
Memory limit: 256 mebibytes

In this problem we will deal with a *hash table with open addressing strategy* — an implementation of “set” data structure based on hashing. Naturally, hash table is concerned with *hash values* of elements, which are non-negative integers computed in a certain way based on the contents of the elements.

The hash table data structure operates on an infinite array with 0-based indexation. Each cell of the array can either be empty or contain an element. We will only consider one type of instruction: *add* an element with hash h . While performing this instruction, we try to put current element in cells $h, h + 1, h + 2, \dots$ in this order. The first empty cell among these is then filled with the element.

We consider the *cost* of the instruction to be the number of occupied cells we skipped while trying to put the element. For example, if an element with hash 5 was immediately put into cell 5, the cost is zero, but if it ended up in cell 10, we must have performed five unsuccessful tries and thus the cost is 5.

Consider a sequence of “add an element with hash h_i ” instructions. We are interested in changing the sequence (inserting or omitting instructions) while maintaining the total cost of all instructions performed in this order. The hash table is cleared before every sequence is processed.

Initially the instruction sequence is empty. You have to process two types of queries changing the sequence:

- “+ $h p$ ”: insert the instruction “add an element with hash h ” so that it results on p -th place in the sequence;
- “- q ”: delete the instruction that was inserted on q -th query.

After performing each query you should print the total cost of all instructions currently in the sequence performed in this order.

Instructions and queries are numbered starting from 1. It is guaranteed that all queries are valid.

Input

First line of input contains a single integer n — the number of queries to the sequence ($1 \leq n \leq 150\,000$).

Next n lines describe the queries. Each of these lines is either formatted as “+ $h p$ ” or “- q ”, describing an insertion or deletion query according to the rules above.

Each insertion query “+ $h p$ ” satisfies $0 \leq h \leq 10^5$ and $1 \leq p \leq k + 1$, where k is the size of the sequence before the query.

For each deletion query “- q ”, it is guaranteed that q -th query was an insertion query and it wasn’t deleted as a result of an earlier deletion query.

Output

Print n numbers, each on a separate line: q -th line should contain the total cost of performing the instruction sequence after q -th query.

Example

standard input	standard output
5	0
+ 1 1	1
+ 1 2	2
+ 2 3	0
- 2	0
- 1	