Fully-Persistent Queue

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given a set of operations and a queue Q, you should process operations in order. Each operation is one of the following types:

- `enqueue x`: add x to the end of the queue.
- `dequeue`: remove the front element of the queue. If the queue is empty, do nothing.
- `sum`: print the sum of the elements of the queue. If the queue is empty, print 0.
- `checkout v`: go to version v of the queue.

The initial version is 0 which represents an empty queue. When a `dequeue` or `enqueue` operation is done, a new version is created and becomes the current version of the queue. New versions are enumerated sequentially starting from 1.

Input

The first line contains a single integer q ($1 \le q \le 10^5$) — the number of operations.

Then q lines follow, each describing an operation as mentioned above. The enqueued elements will be 32-bit signed integers.

Output

For each query of type `sum`, print one line containing the sum of the elements of the queue.

Example

standard input	standard output
12 enqueue 4 enqueue 7 enqueue 10 sum checkout 2 sum dequeue sum checkout 3 dequeue enqueue 20 sum	21 11 7 37

Note

Hint: you can answer all queries offline.

Note that `dequeue` operations always create a new version if nothing is dequeued (the queue was already empty).