Dynamic maximum sum (2) P84977_en

Here, you have to efficiently keep a list of integer numbers, which is initially empty. Let the current list be $x_0, \ldots, x_{n-1}$. There are just two operations:

- Given an integer x and any position j between 0 and n, insert x before the j-th position (at the end, if $j = n$). That is, the new list must be $x_0, \ldots, x_{j-1}, x, x_j, \ldots, x_{n-1}$.
- Report the maximum sum of all the consecutive subsequences of the list.

Input

Input consists of several cases. Every case begins with the number of operations m, followed by the m operations. We have an `M` for reporting the maximum, and `I` x j for inserting. Assume $1 \leq m \leq 2 \cdot 10^5$, $-10^{12} \leq x \leq 10^{12}$, and that j is between 0 and the current list size.

Output

For every case, and for every `M` operation, print the maximum sum of consecutive elements inside the current list. Print a line with 10 dashes at the end of each case.

Sample input 1

```
8
I 5 0
M
I 1 1
M
I -3 1
M
I 4 2
M

3
M
I -100 0
M

6
I 1000000000000 0
I 1000000000000 0
I -1 1
M
I 1000000000000 3
M
```

Sample output 1

```
5
6
5
7
----------
0
0
----------
1999999999999
2999999999999
----------
```

Problem information

Author: Salvador Roura

Generation: 2026-01-25T12:04:45.264Z