# Practice Problem B : Stacks of Coke


I'm a programmer, so I drink a lot of coke. To amuse myself, I place the empty cans in a series of stacks, all in a straight line. When I view this series of stacks from the front, I can't always see all the cans, or even all of the stacks, because sometimes a stack can be entirely obscured by a larger stack.

When I view the series of stacks from the front, I can **infer** a minimum number of total cans. I can see each stack that is strictly larger than all the stacks between it and the front of the structure. For each stack I can see, I know how many cans are used to construct that stack. I then total up the number of cans that I know must exist: let's call this number $A$. I then say I **infer** $A$ cans in this structure.

Let's look at an example. Suppose a series of stacks has the following stack sizes, in order from the front to the back: $\{1, 4, 3, 4, 6, 6, 2\}$. I'm able to **infer** 11 cans from this series: I can see the first, second, and fifth stacks (with 1, 4 and 6 cans respectively). I can't see either the third or fourth stack (of size 3 and 4) because they are obscured by the second stack, and I can't see the last two stacks (of size 6 and 2) because the fifth stack is in the way.

Can you write a program that takes a series of stacks, and tells me how many cans I can **infer** ?

**Input Specification:**

The input consists of a series of test cases, one per line. Each line begins with an integer $N$, the number of stacks, followed by $N$ integers representing the sizes of the stacks from front to back. The input ends on EOF. There are never more than 50 stacks of coke, and no stack has more than 50 cans.

**Output Specification:**

For each test case output a single number: the number of cans that I can **infer**.

**Sample Input:**

```
7 1 4 3 4 6 6 2
```

**Sample Output:**

```
11
```

*Andrew Henrey*