Problem F - 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. 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.

At the beginning of the day, I am able to infer $A$ cans. During the day, I think I drank $C$ cans of coke, and placed the empties into the series somewhere (perhaps at the end). Note that the only change I’ve made to the structure of the stacks is to place these $C$ cans, I don’t re-arrange any cans that were already there. At the end of the day, I look at my series of cans and I can infer a different number of cans, $B$. Amusingly, $B$ is actually less than $A$.

My memory is pretty bad, so I might have been wrong about $C$. Please tell me whether there was some arrangement of cans where I could infer $A$ cans, place $C$ cans, and then infer $B$.

Input Specification:

The input consists of a series of test cases, one per line. On each line you are given three positive integers, $A \leq 50$, $B < A$, and $C \leq 10$. The input ends on EOF.

Output Specification:

For each test case, output Possible if there is a legitimate configuration of the cans, and Impossible otherwise.

Sample Input:

```
10 4 3
43 31 1
16 5 4
```

Sample Output:

```
Possible
Possible
Impossible
```