Reasoning Table for Do_Nothing_Realiz for Integer_Template

```
Operation Do_Nothing(i: Integer);
   requires i + 1 <= max_int;
   ensures i = #i;
```

State	Code	Assume	Confirm
0		Precondition for Do_Nothing $\mathcal{P}1: i_o + 1 \leq max\_int$	$C1: i_o + 1 \leq max\_int$
	`Increment(i)`		
1		$\mathcal{P}2: i_1 = i_o + 1$	$C2: min\_int \leq i_1 - 1$
	`Decrement(i)`		
2		$\mathcal{P}3: i_2 = i_1 - 1$	*Postcondition for Do_Nothing* $C3: i_2 = i_o$

Reference:

`Operation Increment(i: Integer);`
`  requires i + 1 <= max_int;`
`  ensures i = #i + 1;`
	`Operation Decrement(i: Integer);`
	`  requires min_int <= i - 1;`
	`  ensures i = #i - 1;`
		`Constraint from Integer_Template`
		`  min_int <= i <= max_int`

VCs for States 0, 1, and 2:

VC0: $(\mathcal{P}1) \rightarrow C1$

VC1: $(\mathcal{P}1 \wedge \mathcal{P}2) \rightarrow C2$

VC2: $(\mathcal{P}1 \wedge \mathcal{P}2 \wedge \mathcal{P}3) \rightarrow C3$

Proof for VC0, VC1, and VC2

VC0: $(i_o + 1 \leq max\_int) \rightarrow (i_o + 1 \leq max\_int)$

$True$

VC1: $((i_o + 1 \leq max\_int) \wedge (i_1 = i_o + 1)) \rightarrow (min\_int \leq i_1 - 1)$

$(min\_int \leq i_o)$(because of Integer template constraint $min\_int \leq integer \leq max\_int$)

$\rightarrow (min\_int < i_o + 1)$

$\rightarrow (min\_int < i_1)$ (substitution using $i_1 = i_o + 1$)

$\rightarrow (min\_int \leq i_1 - 1)$

VC2: $((i_0 + 1 \leq max\_int) \wedge (i_1 = i_0 + 1) \wedge (i_2 = i_1 - 1)) \rightarrow (i_2 = i_0)$

$\rightarrow$ $(i_0 = i_1 - 1)$ *(because* $i_1 = i_0 + 1$*, and subtract 1 from both sides)*

$\rightarrow$ $(i_0 = i_1 - 1 = i_2)$ *(because* $i_2 = i_1 - 1$*)*

$\therefore$ $(i_0 = i_2)$ *(simplification)*

□