Reasoning Table for removeTwo

Two additional aspects to filling in a Reasoning Table

To Do:

1. Fill in the reasoning table slots A0, A1, A2, C0, C1, C2 with the correct assertions
2. In the assertions, substitute self for the controlling object and the actual parameter for the formal parameter
3. In the assertions, postpend to the variable its correct state number

```
void removeTwo(SequenceOfT& s, T& z, T& y);
   //! updates s
   //! replaces z, y
   //! requires   |s| > 1
   //! ensures    s = #s[2,|#s|)  and  <z> is prefix #s  and  <y> is prefix #s[1,|#s|)
```

State	Code	Assume	Confirm
0		*A0*: `|s0| > 1`	*C0*: `0 <= 1 < |s0|`
	`s.remove(1,y);`		
1		*A1*: `<y1> = s0[1,2) and s1 = s0[0,1 * s0[2,|s0|) and z1 = z0`	*C1*: `0 <= 0 < |s1|`
	`s.remove(0,z);`		
2		*A2*: `<z2> = s1[0,1) and s2 = s1[0,0) * s1[1,|s1|) and y2 = y1`	*C2*: `S2 = s0[2,|s0|) and <z2> is prefix s0 and <y2> is prefix s0[1,|s0|)`

1) When the actual parameter is a literal just replace formal parameter with the literal in the assertion

2) When a variable's value remains the same from one state to the next, you have to note that as in these two examples

Reference:

```
   void remove (Integer pos, T& x);
      //! updates self
      //! restores pos
      //! replaces x
      //! requires: 0 ≤ pos < |self|
      //! ensures: <x> = #self[pos, pos+1) and self = #self[0, pos) * #self[pos+1, |#self|)
```

Reminder: Each and every variable appearing in each of the assertions found in an RT must have the correct state number postpended to it.
In this RT, variables *s*, *z*, *y* all have state #s in every cell