

The end of the for loop's body triggers the increment of its control variable *i*, so it now has the value 2

*i* is still in the range 0..2 so it runs the body again

2

Function: Sum

-----

Returns: Double - The sum of the numbers from the data array

Parameters:

- 1: data (by const ref, array of Double) - values to sum
- 2: size (Integer) - the number of elements in data (C only)

Local Variables:

- \*: *i* (Integer) - index of the current element in the array
- \*: total (Double) - running total

Steps:

- 1: total is assigned 0
- 2: For *i*, starts at 0 and loops to the highest index of data
- 3: total is assigned total + data[*i*]
- 4: Return the result, total

Procedure: Main

-----

Local Variables:

- \*: my\_data (array containing 3 Double values) - data array

Steps:

- 1: Call Populate Array ( my\_data, 3 )
- 2: Output 'Sum is ', and Sum ( my\_data
- 3: ..

3

1

This is the end of the loop, so *i* must be incremented and control returns to step 2

Sum	
data[]:	10.0
size:	3
i:	2
total:	5
result:	92
Instruction:	Step 2

Main	
my_data[]:	10.0
	-5
	17.21
Instruction:	Step 2

