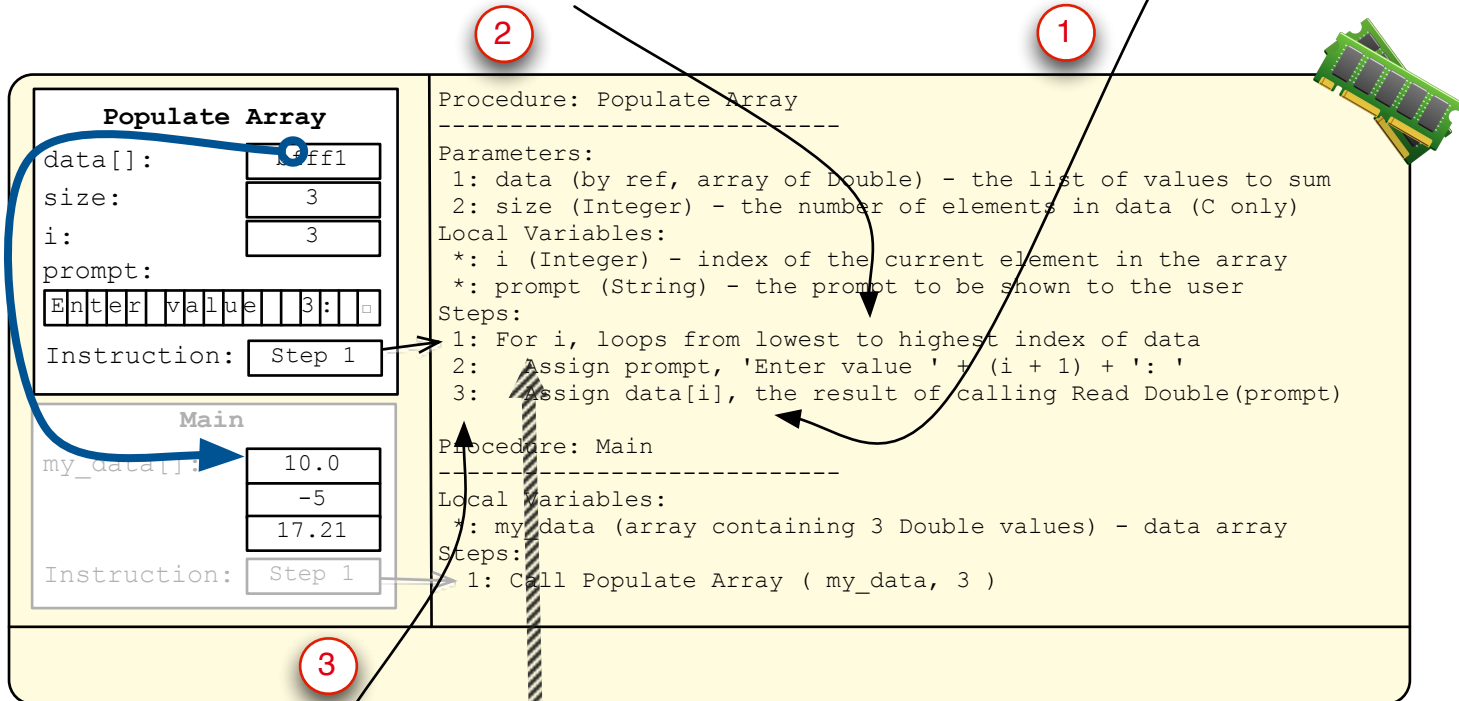The loop checks its condition to see if it is still in range (0 .. 2), as it is not the loop's body is skipped

②

This is the end of the loop body, so i is incremented to 3, and control jumps back to step 1

①

Populate Array

data[]:	0xff1
size:	3
i:	3
prompt:	
Enter value 3: □	
Instruction:	Step 1

Main

my_data[]:	10.0
	-5
	17.21
Instruction:	Step 1

```
Procedure: Populate Array
-------------------------
Parameters:
 1: data (by ref, array of Double) - the list of 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
 *: prompt (String) - the prompt to be shown to the user
Steps:
 1: For i, loops from lowest to highest index of data
 2:    Assign prompt, 'Enter value ' + (i + 1) + ': '
 3:    Assign data[i], the result of calling Read Double(prompt)

Procedure: Main
-------------------------
Local Variables:
 *: my_data (array containing 3 Double values) - data array
Steps:
 1: Call Populate Array ( my_data, 3 )
```

③

No steps follow the loop, so after this Populate Array will end

CPU

Terminal — bash — 80×24

Enter value 1: Ten
Please enter a number.
Enter value 1: 10.0
Enter value 2: -5
Enter value 3: 17.21