

Finally 5.0 is stored in total

3

The value of total is read, it is 0 at this time. So the expression  $total + data[i]$  has the value 5.0

4

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: ...

Sum

data[]:	ref1
size:	3
i:	1
total:	5.0
result:	92
Instruction:	Step 3

Main

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

2

1

Reads the  $i^{\text{th}}$  element of the data array - i is 1, so data[1]

data[1], skips 1 element to find its value - reading -5



```
Terminal — bash — 80x24
Enter value 1: Ten
Please enter a number.
Enter value 1: 10.0
Enter value 2: -5
Enter value 3: 17.21
```

