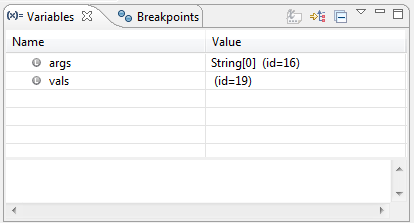
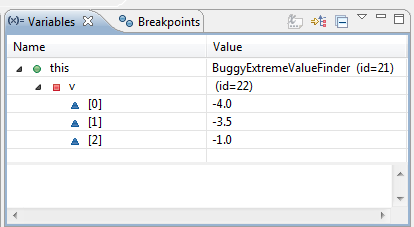
CS162, Lab 1

### Section 1, ExtremeValueFinder

# Q1

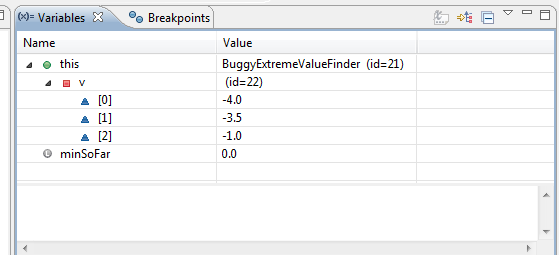


# Q2

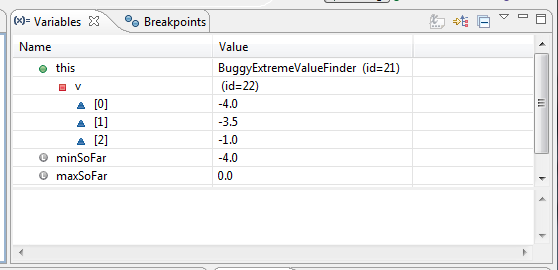


# Q3

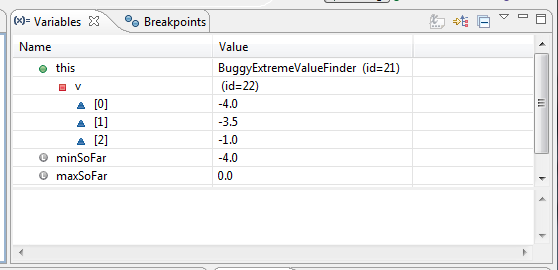
## Before Iteration



## Iteraton 1; min = -4, max = 0



## Iteration 2; no change



# Q4

Loop executed 2 times.

# Q5

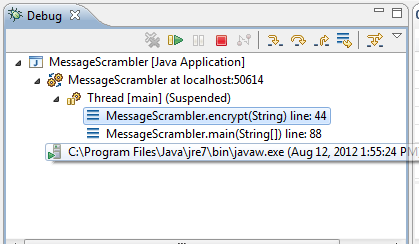
First, I had to correct the length of the iteration loop from **for**( **int** i = 0; i < (**this**.v.length-1); i++ ) to **for**( **int** i = 0; i < (**this**.v.length); i++ ). This way it executes over every element in the array.

Also, it checks for the maximum against zero, which isn’t necessarily the max of the array, if all the numbers are negative. So, I changed the max code to check agains the absolute value of the array element, like so:

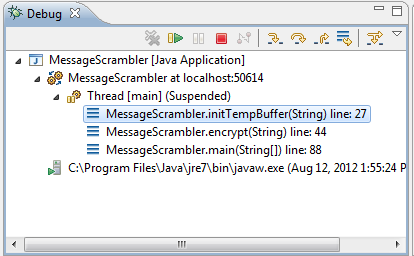
**if**( Math.*abs*(**this**.v[i]) > maxSoFar ) maxSoFar = **this**.v[i];

### Section 2, the message scrambler

# Q1



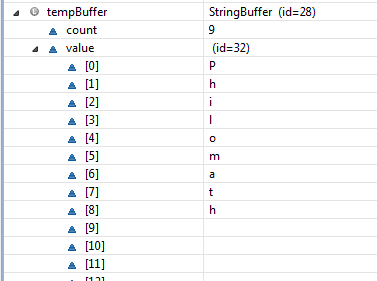
# Q2



We were in the MessageScrambler.main() function and it called MessageScrambler.encrypt(), now we are looking at the functions inside that.

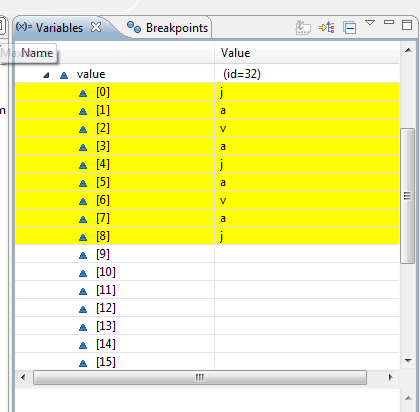
# Q3

Philomath



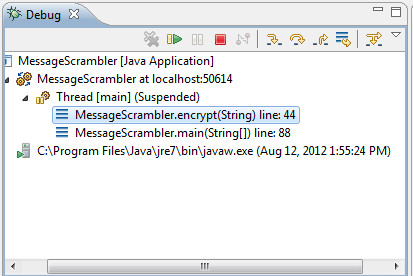
# Q4

Javajavaj



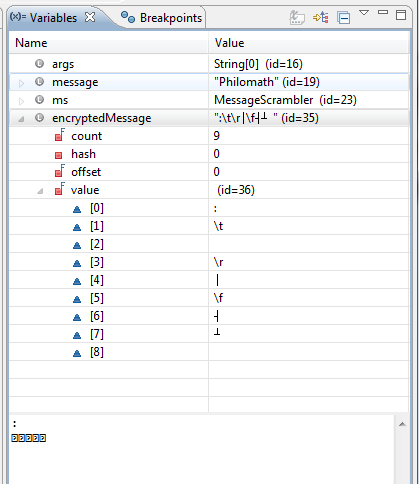
# Q5

We exited that function back to the function that called it. Thus the call stack shows the previous function



# Q6

:\t­\r\f



# Q7

Philomath

