Syntactic Bug (before fix):

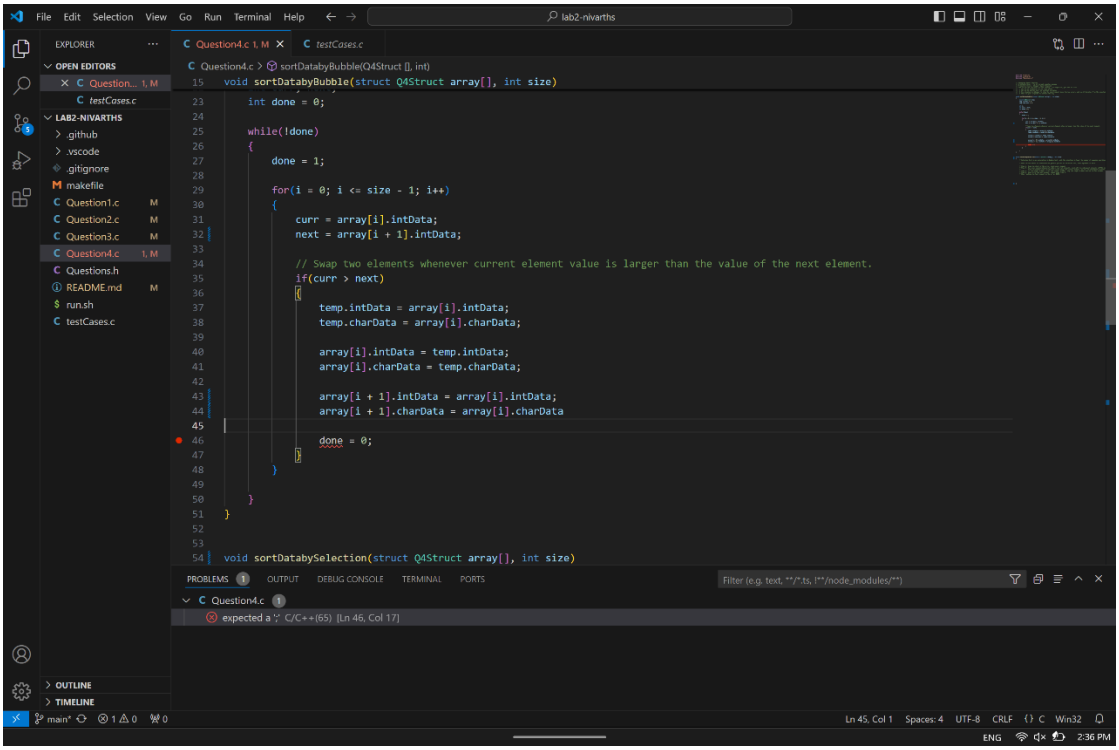

Figure 1: Compiler shows missing ";" at line 44.

Syntactic bug validation (after fix):

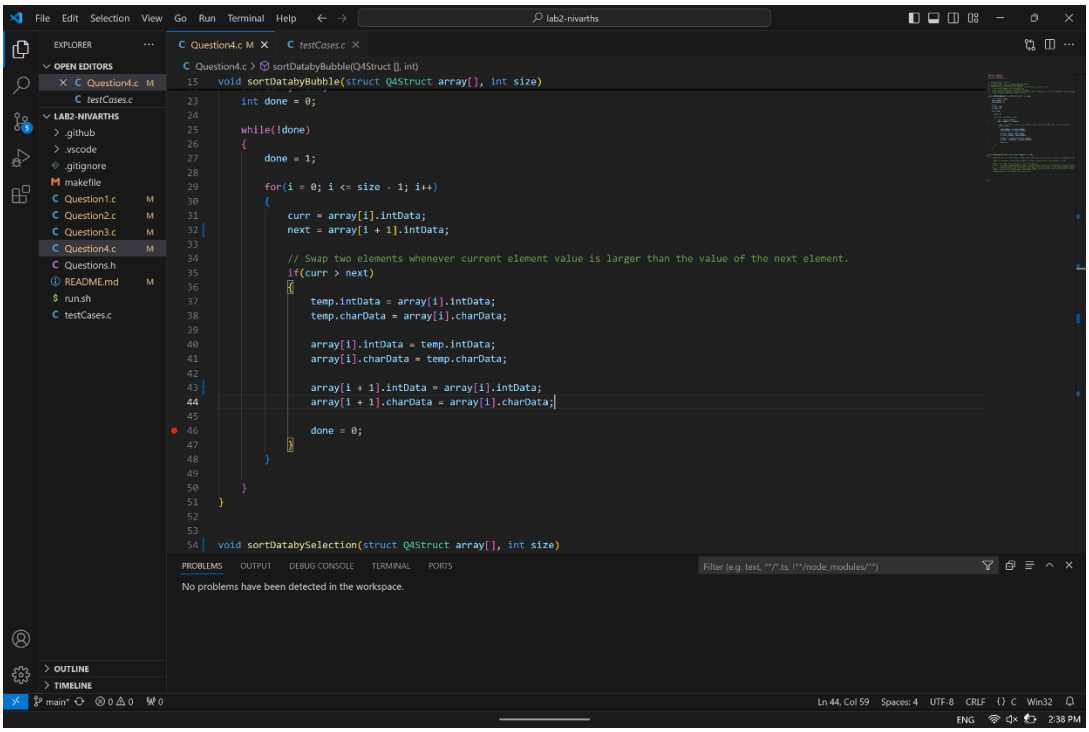

Figure 2: Added ";", compiler shows no error.

Semantic Bug #1:

Program does not swap the current and next elements of the array as intended, keeping the values where they initially were. Using the debugger, breakpoint was placed at line 46, and program was carried out using 'step into' and 'step over'. Watching the variables array[i] and array[i+1], the debugger indicates that the current value is being reassigned to the same value instead of being reassigned to the next term.

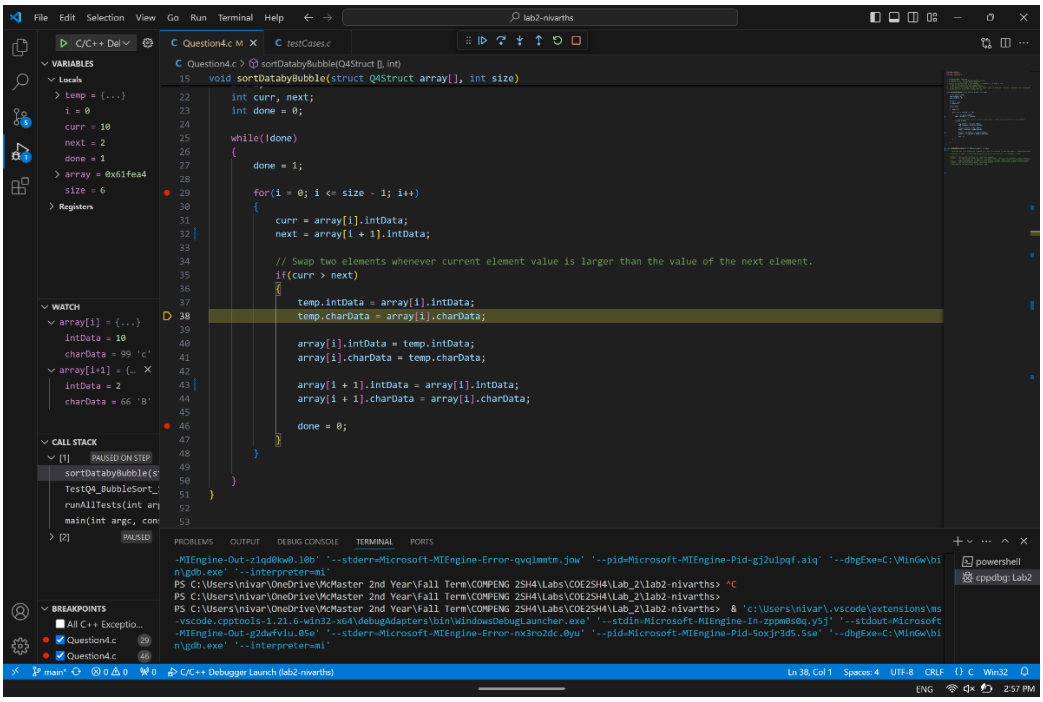

Figure 3: Pre swap before debug

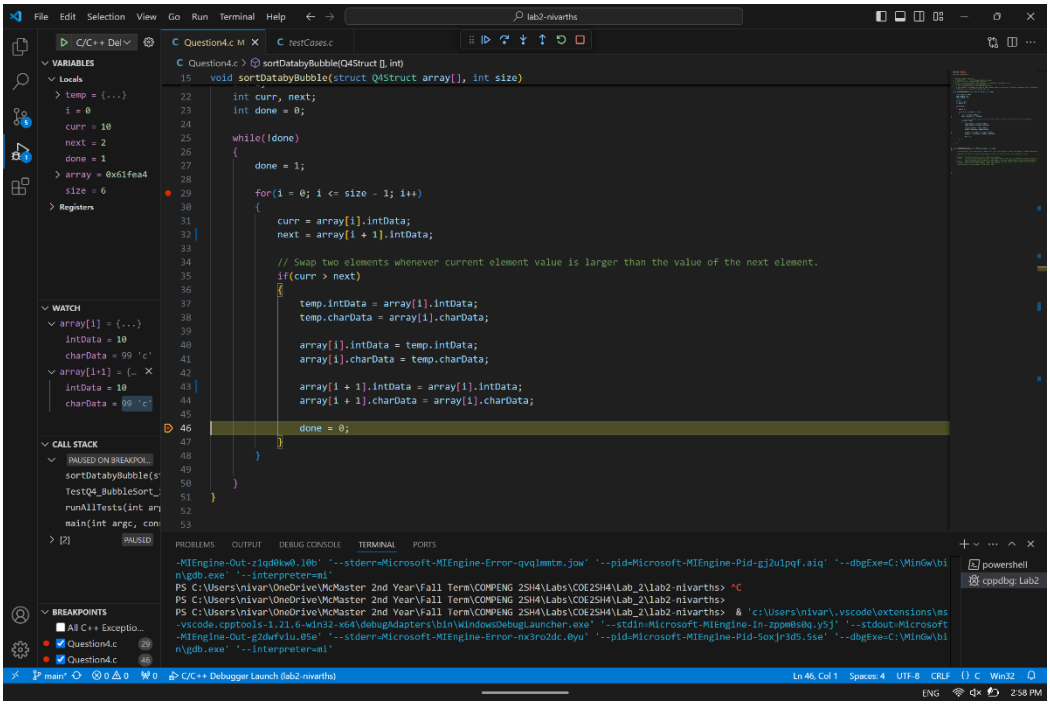

Figure 4: Post swap before debug

The problem may be from improper variable assignment. Lines 40-41 show array[i] assigned to the temp variable, when it should be assigned to the next term (array{i+1]). As well, lines 43-44 show that array[i+1] is assigned to array[i], when it should be assigned to temp.

Bug Validation

Lines 40-41 have been assigned to the next term and lines 40-43 have been assigned to the temp variable, resulting in swapping.

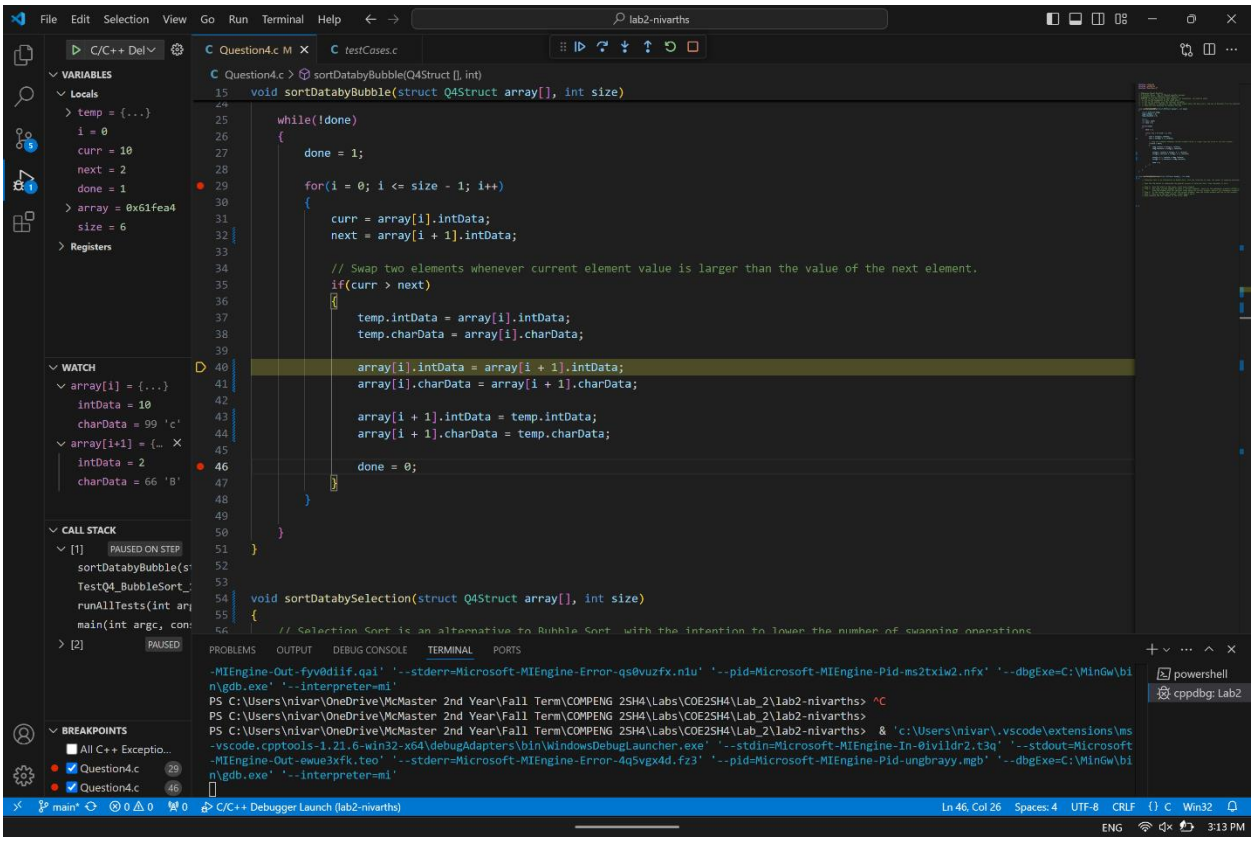

Figure 5: pre swap after debug

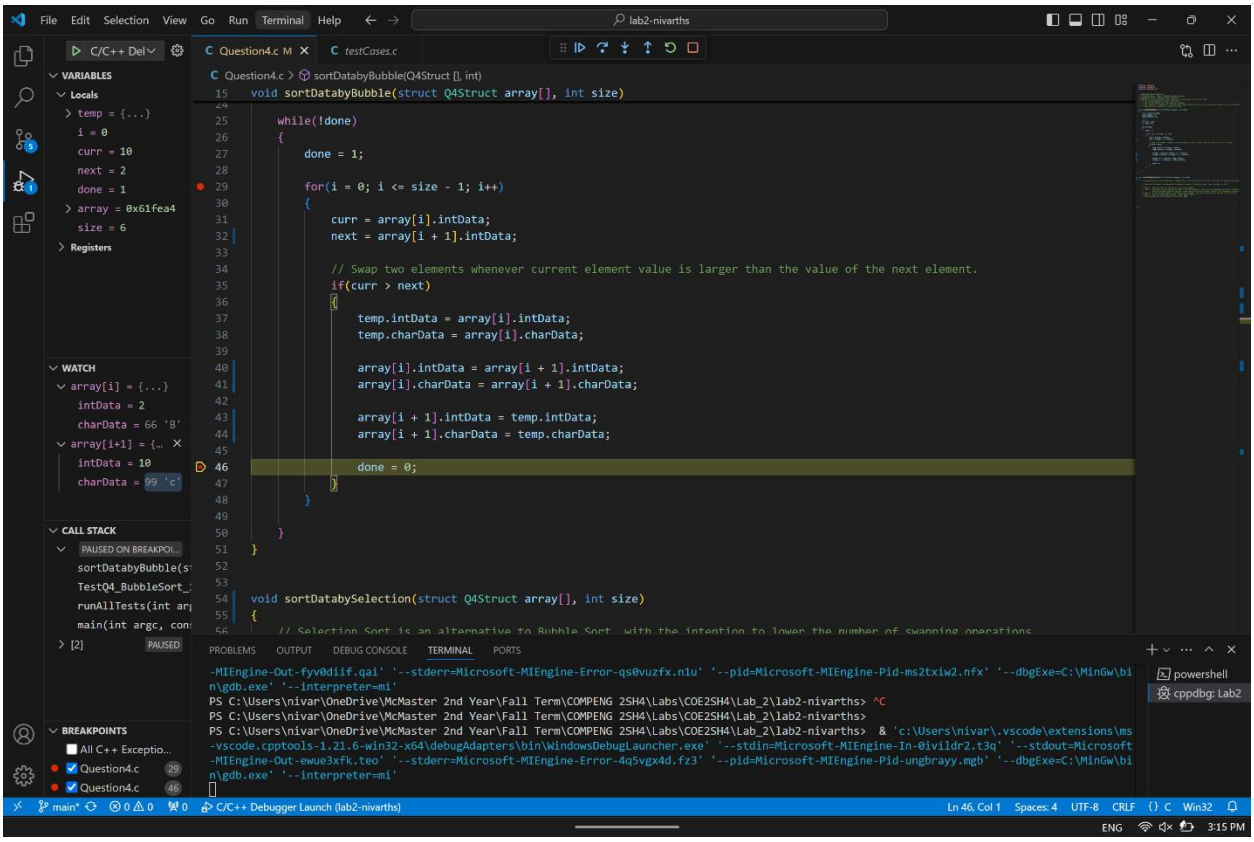

Figure 6: post swap after debug

Semantic Bug #2:

Program runs ‘out of boundaries’ on last iteration of for loop on line 29, accessing ‘garbage’ contents in memory. Size is defined to be 6, meaning that indices of array go from 0-5. Variable ‘i’ should stop at 4 to account for array[i+1], but resets and exits loop after i = 5.

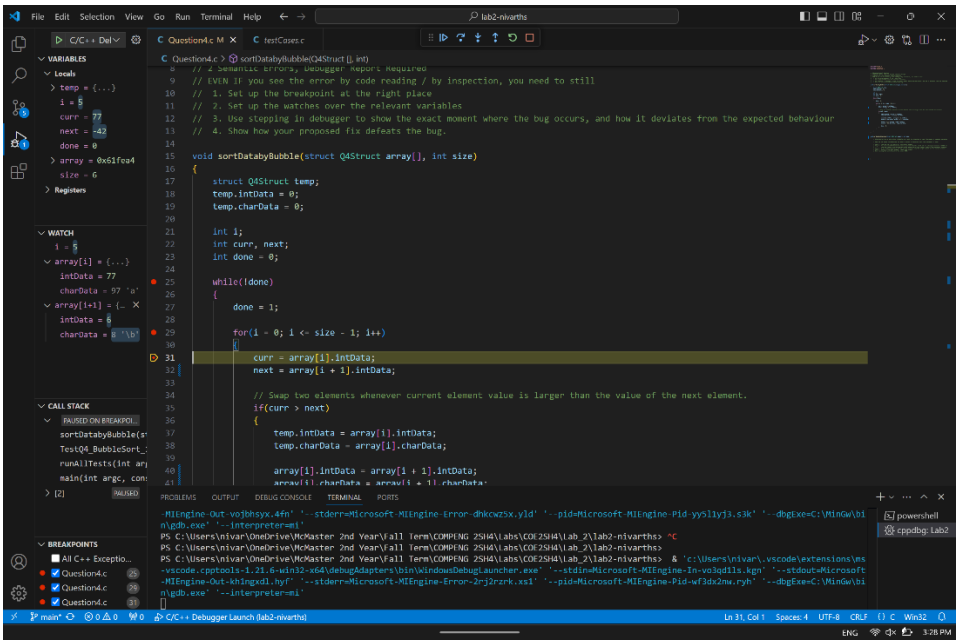

Figure 7: before debug (still in loop)

Using debugger and placing breakpoints at lines 29 and 31, possible root cause was discovered after iterating. Program was still running in the loop after i = 5. Line 29 shows loop to iterate from i = 0 to i <= size – 1, indicating that there would be no room for array[i+1], thus accessing out of bounds memory.

Validation: range was changed from i <= size – 1 to i < size – 1, leaving room for array[i+1] to access values in array

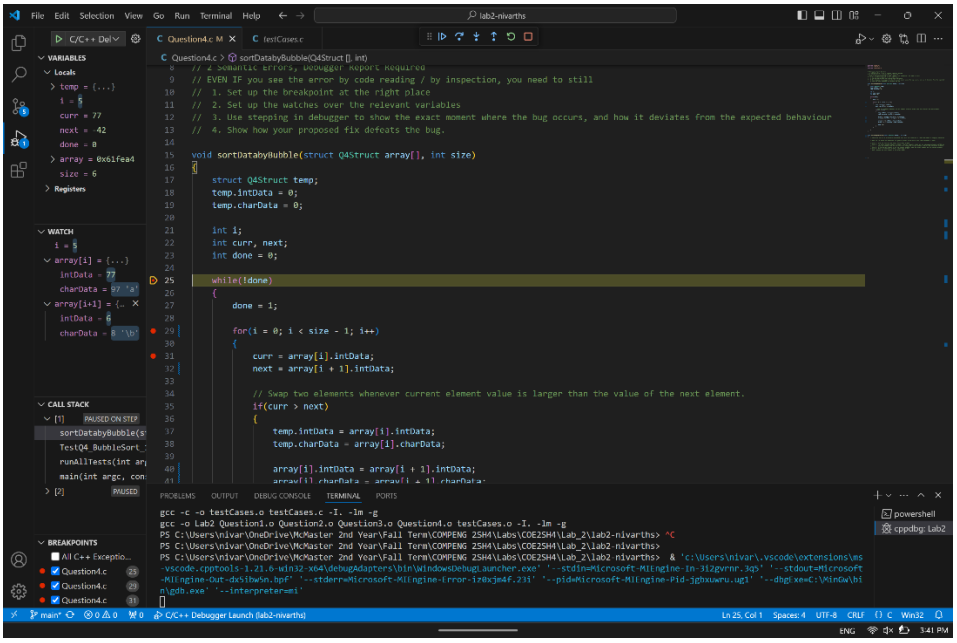

Figure 8: after debug (exits loop)