Semantic Error 1:

Program would not swap the minIndex value to the character with the lowest ASCII value, and instead would continue to compare the first value to the rest.

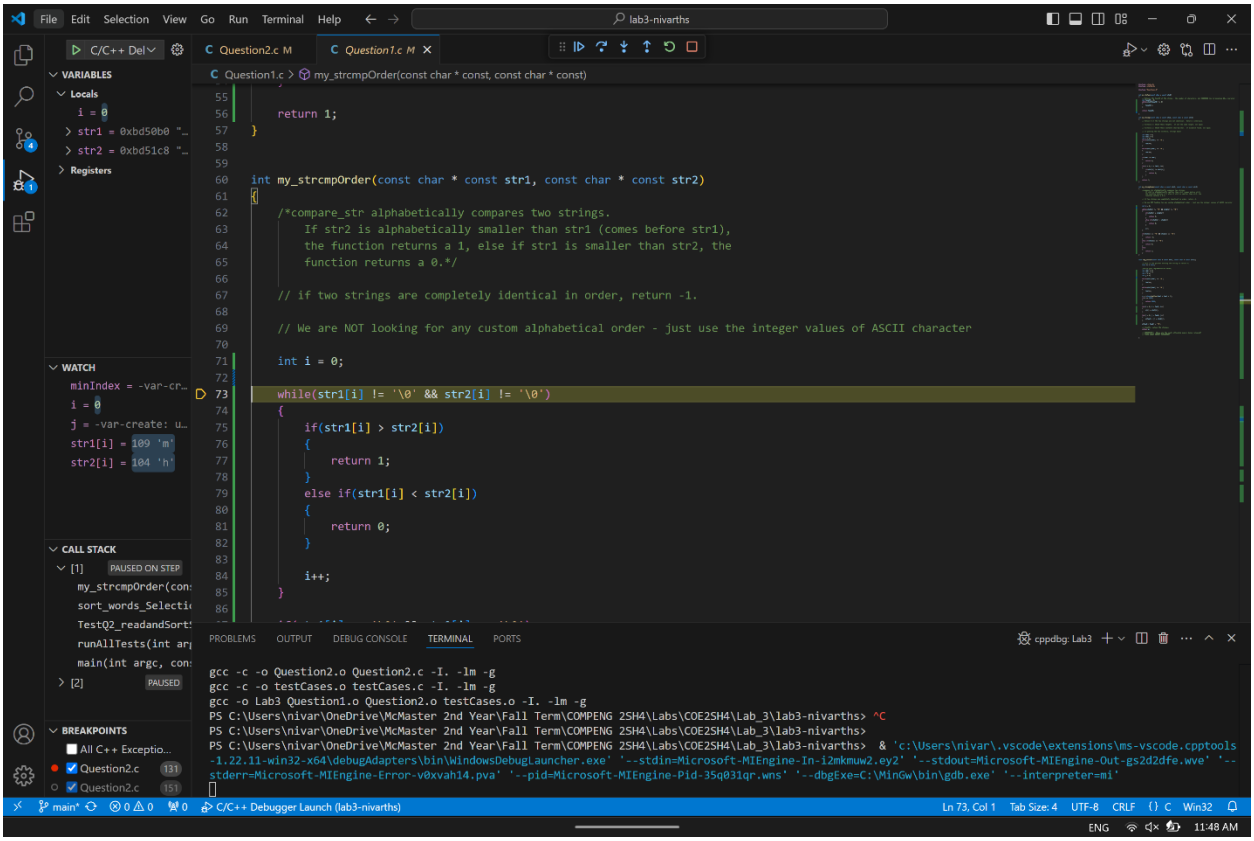

Figure 1: program before swap before debug, characters compared are ‘m’ and ‘h’.

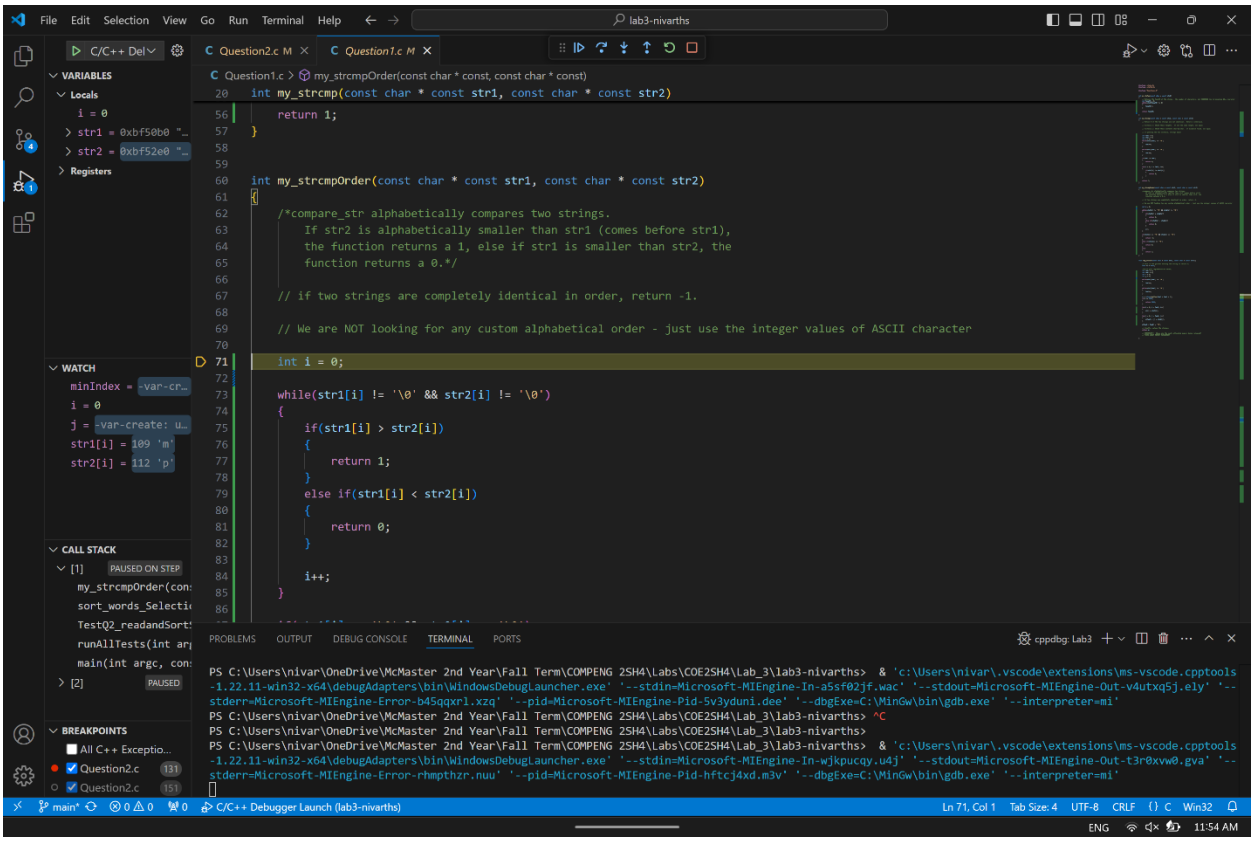

Figure 2: program after swap before fix, characters being compared are ‘m’ and ‘p’

The above figure shows the error in the program. The ASCII value for ‘m’ is higher than ‘h’, so ‘h’ should be set to minIndex as it is the lowest value currently in the program loop. However, ‘m’ continues to be compared.

The error resides in line 137, where the first argument in the function is words[i]. This is incorrect as throughout the first iteration of the loop, the first character will be compared. Instead, we want it to be words[minIndex], so every comparison will have the minimum value compared.

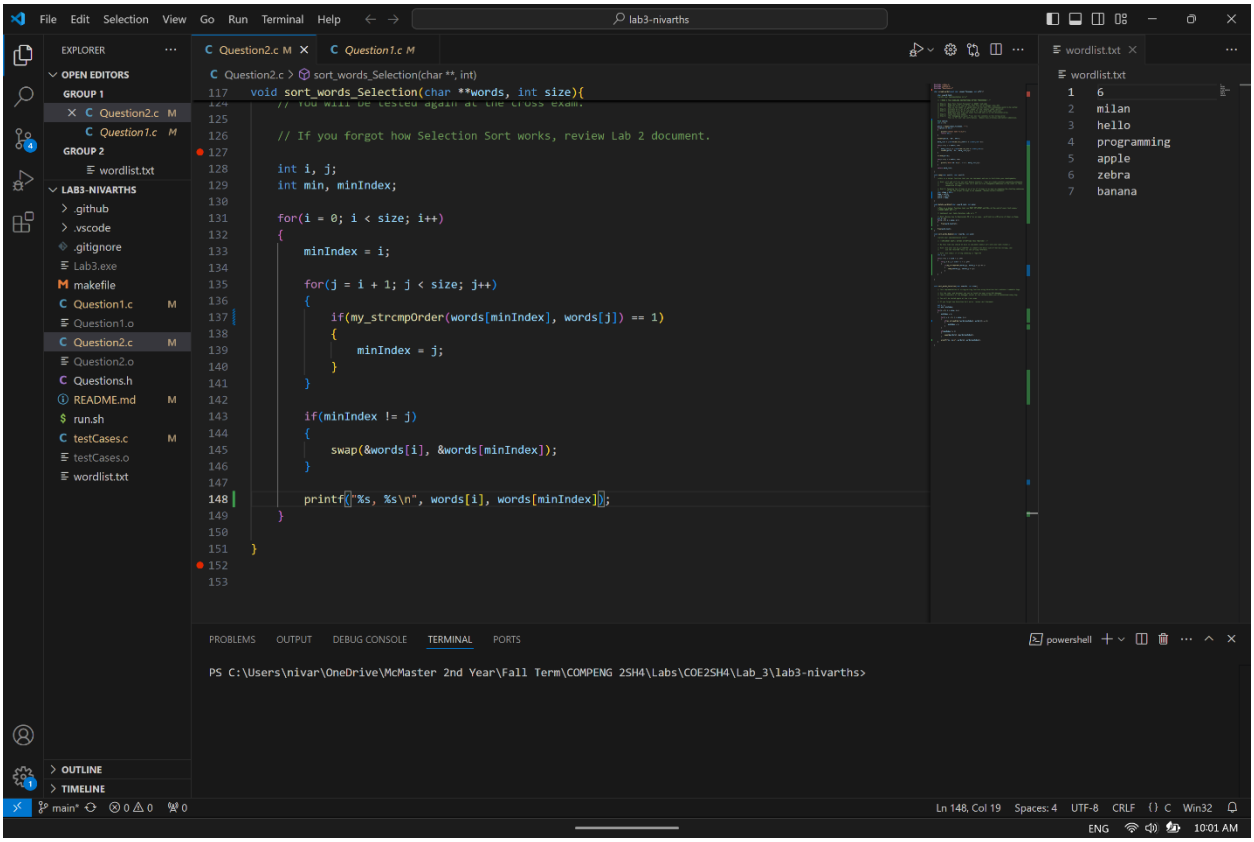

Figure 3: Bug fix validation, words[i] was replaced with words[minIndex].

The program now runs as intended.

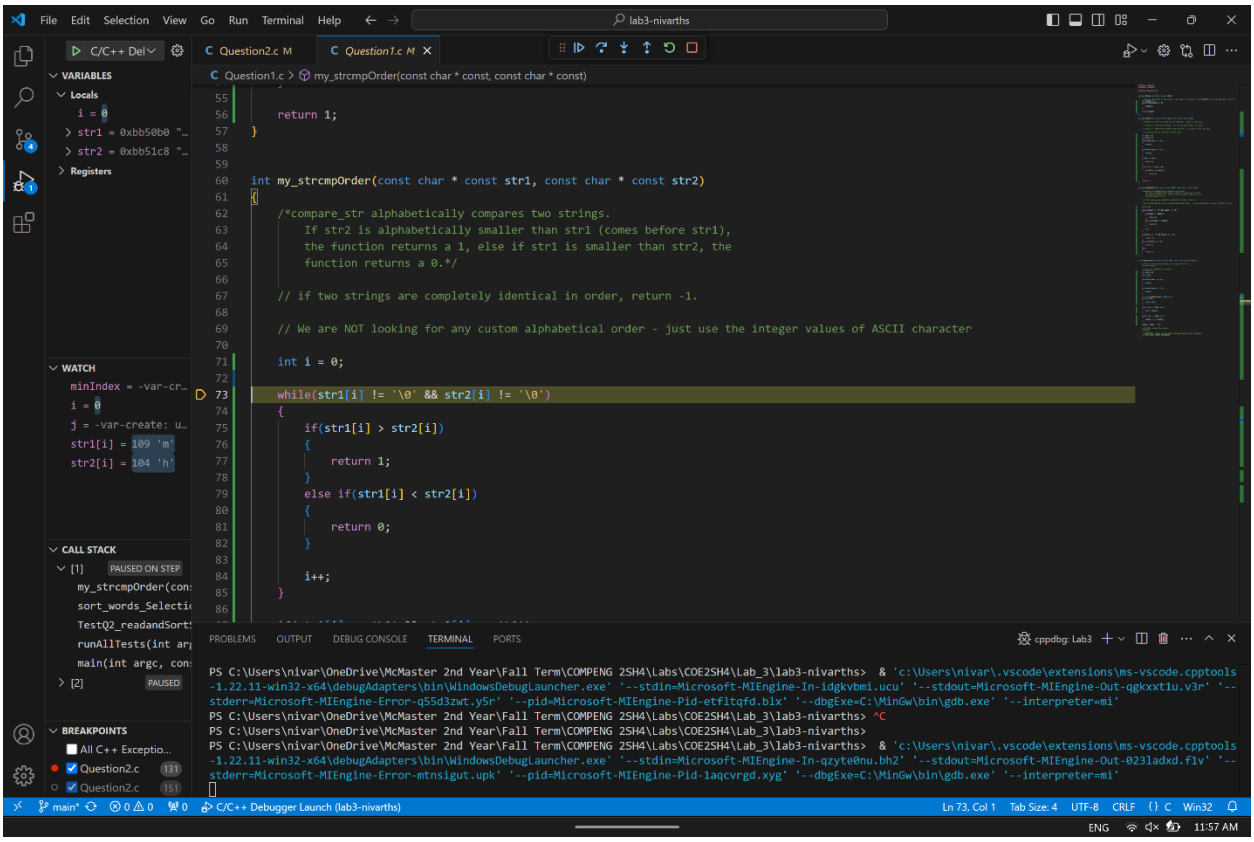

Figure 4: program before the swap but after debug

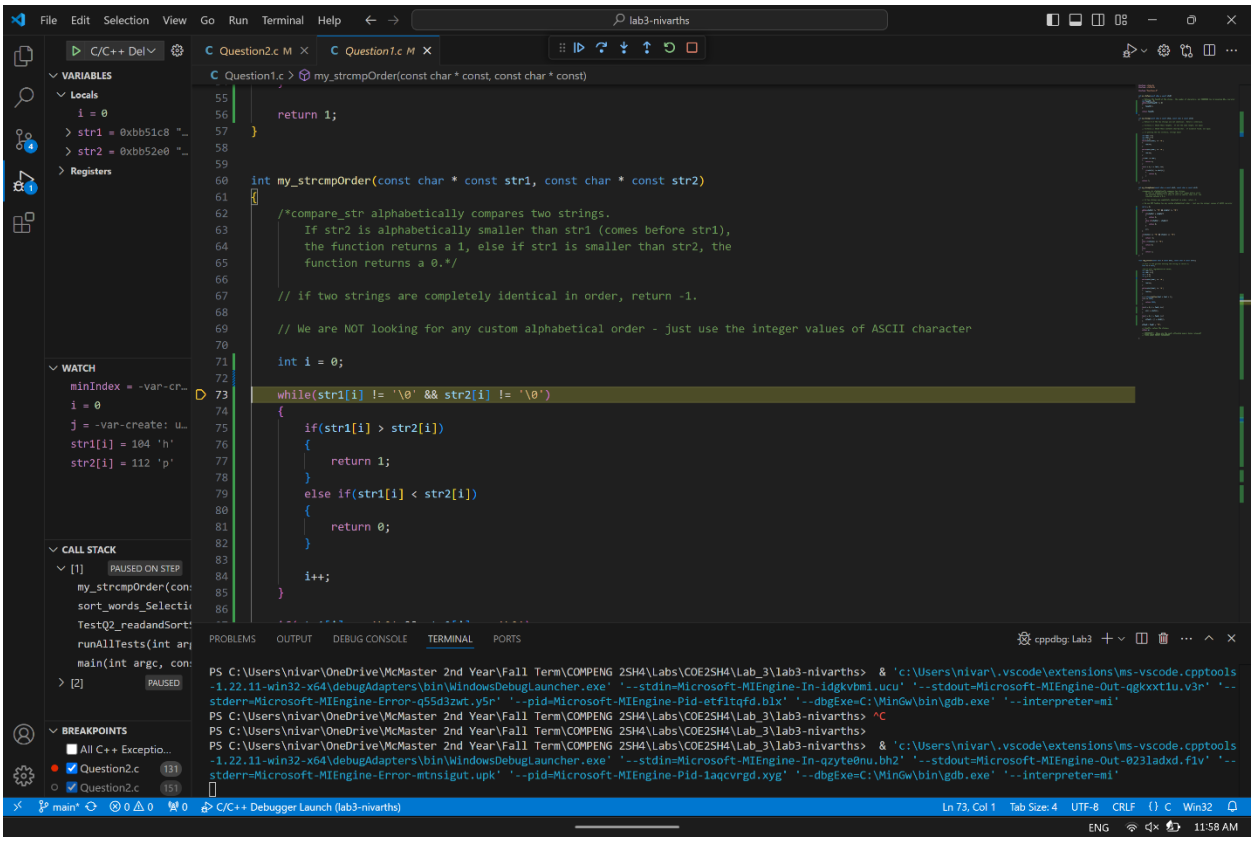

Figure 5: program after swapping and debugging, showing the proper comparison between the characters.

Semantic error 2:

Line 143 states what if minIndex != j, then the swap will occur. This is incorrect as the swap shouldn't occur when the minimum ASCII value isn't i, and a smaller value has been found.

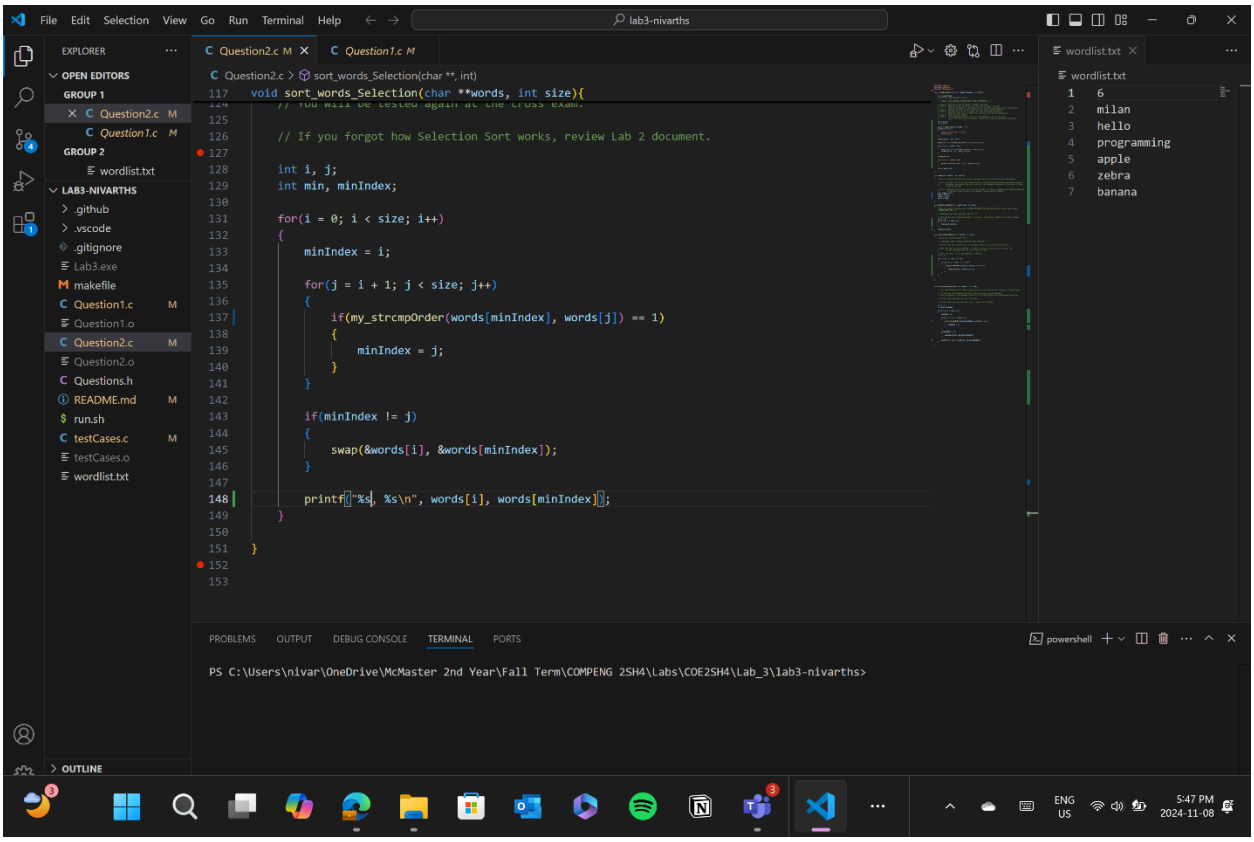

Figure 6: semantic error 2, showing minIndex != j.

This error does not affect the code as the program still runs and sorts the list of words correctly. However, if a different set of words was provided, the swap may not have occurred correctly, as a minimum value may not be found.

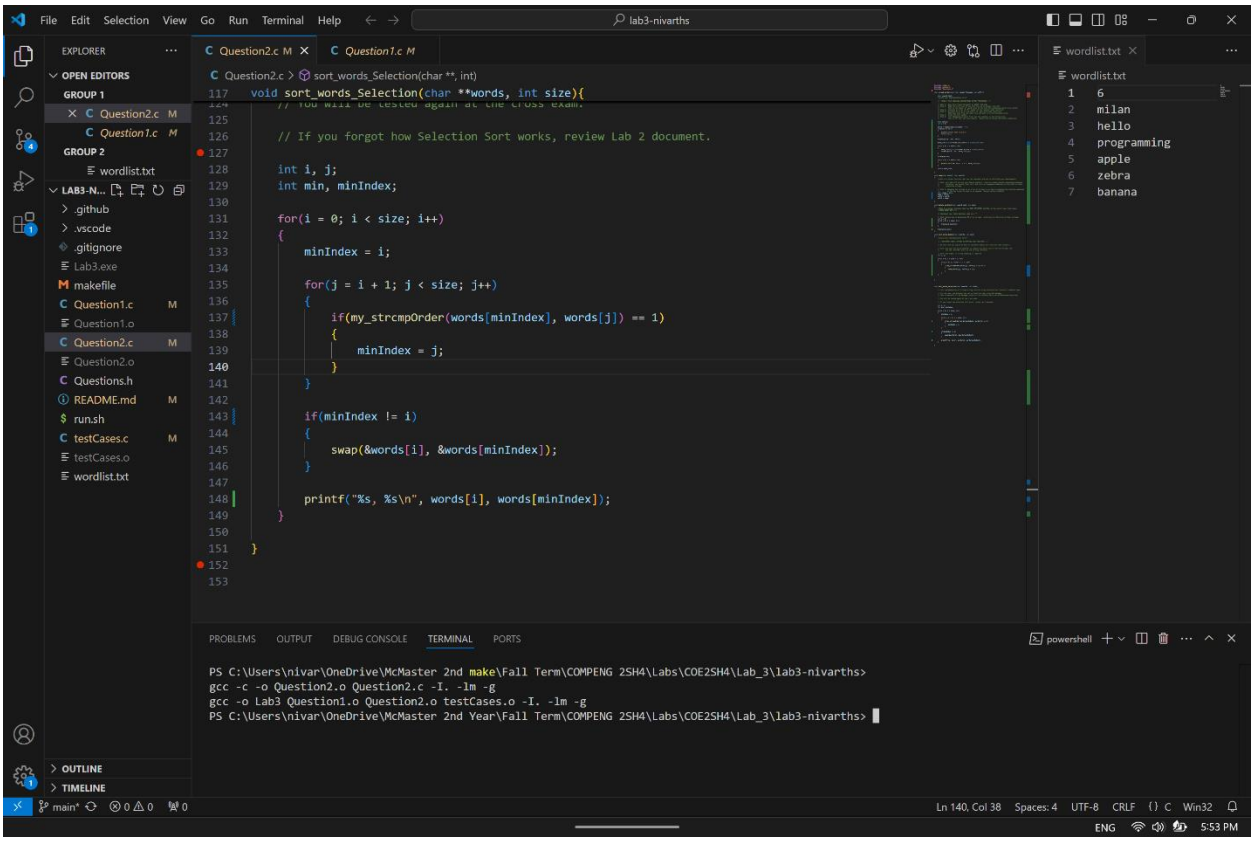

Figure 7: semantic error fixed, showing minIndex != i.