Assignment 7 -- Group Part

Robert Brempelis
Andrew Schapper
Hailey Wilder
Michael Ross

1. Which program did your group decide on?

The overall structure in all codes was similar (although order varied in some cases), following the pattern of initializing variables, looping through the integers and counting the occurrences to find the max values by incrementing a counter each time a match to the current integer is found, clearing the counter if/when a new max is found, and sorting the mode value.

We chose to use the code written by Hailey Wilder because it was used the fewest number of loops (providing efficiency), responded within the scope of the prompt (passed all test scenarios provided below), and was commented appropriately (comment blocks and line lengths considered).

Test Scenarios: Condition Input Array Output

Single mode {1, 2, 3, 3, 5} 3

Two modes {1, 2, 2, 3, 3} 2, 3

Three modes, not sorted {6, 6, 1, 1, 4, 4} 1, 4, 6

All numbers are mode {5, 1, 3, 2, 4, 7, 6} 1, 2, 3, 4, 5, 6, 7

Other test1 {2, 2, 4, 4, 8, 4, 8} 4

Other test2 {1} 1

Other test3 {-1, 6, -1, 10, 13, 13} -1, 13

2. What advantages do you think that program has over the others? (be detailed)

The program we choose (Hailey's) was efficient in that it used less loops and calculations than other codes to get the same result. It was also well-annotated, using formatted comment blocks required for this class and in-line comments in order to explain the progression of the analysis. The code also considered line-lengths in order to assure easy reading.

Michael's code gave the same result as Hailey's during test scenario input and return and was very similar structurally. However, more loops were used, which as shown, did not produce an incorrect result, but with larger test cases this can lead to a slower function. There was also an unneeded index variable which would potentially slow down this code more. In small tests there is no noticeable difference, though if this was used on a large scale or called iteratively, there would be noticeable lag. Thus,this function was less efficient than Hailey's.

Andrew's program passed all of the validation checks and had great validation checks. There were some naming convention and assignment issues and some aspects of this program went beyond the scope of what the assignment asked required. Lastly, while each function had a general comment, the flow of what was happening within the functions was not explained with commenting. These ideas all have great potential, but varied from the assignment goals enough that we went with Hailey's program.

Robert’s program had similar structure, but it did not produce the correct mode for all test scenarios, so it still needs a bit of tweaking.

3. What improvements do you think could be made to that program? (be detailed)

In order to improve the chosen code (Hailey’s) it would ideal to begin incorporating validation checks such as those seen in Andrew’s program. While implementing these, it would be important to stay true to the nature of the assignment requirements and not deviate from required naming/assignment values.

Validation checks could include things like checking that an integer was within a specific range or making sure that we are only testing certain types of values (ints) in the code and if not telling the user to enter the appropriate value types.

Finally, having a commented out main would have made testing by others easier, rather than just using a separate test code, despite the fact that either option works.