Assignment 7 Comparisons

Byron Kooima

CS/161

March 1, 2017

Tim Alcon

Assignment 7 Comparisons (Group 13)

Comparison 1 (Eric Buss):

The first program for comparison was Eric Buss'. His program was very similar to my own and followed the sequence similar to Prof Alcon's direction. He structured both for loops in the findMode function exactly like my own function. Eric's program was also picked as the most complete code by the group with only minor highlights for areas of improvement.

One area where I believe my program was better was in the readability of the code. While not required (and functionally equivalent), I initialized all of my variables at the beginning of the function. It wasn't necessary to initialize the counters used in the for loops, but many of the team members commented that it concisely captured the variables used in the function. I also named my array variables more consistently throughout the program. I identified the variables used in my arrays with an "Arr" naming convention. This allowed me to quickly trace which variables were arrays and which were simple integers.

The main area where Eric's program was better than my own was the commenting throughout the code. Before each of his for loop iterations, he included very descriptive comments relating to their function. The group agreed that his was the most descriptive code and allowed everyone in the team to quickly identify what every line was attempting to accomplish.

Comparison 2 (Samuel Gafford):

The second program for comparison was Samuel Gafford's. His program did not function and Sam was unable to upload a working code for the group. He did provide his pseudocode but did not provide anything to compare against.

Besides the obvious area where my program was better, I also believe that my code followed the best for loop construction. Sam's pseudocode stated that he was iterating through

the loop to capture every element of the array followed by an initialization of the array. Based on his layout of the pseudocode, I don't believe the program would have functioned as he intended. I also believe my code was more efficient than initializing two arrays to determine the highest frequency. I originally followed a similar path where I assigned the frequency count to another array, but this created inefficiencies in determining the highest frequency integer.

One area that I did think worked well with two arrays was the ability to quickly push a value from the frequency array to the vector. I found this to be a good way to step through the code and output specific integers and the associated counts while I was debugging my issues.

Comparison 3 (Alexander Goodman):

The third program for comparison was Alexander Goodman's. Alexander took a slightly different approach to the problem and created a single for loop to populate his vector. The team agreed that his code was the most original, but it lacked comments and explanations in his approach.

The main area where I believe my program was better was in readability of the code. I had comments throughout the code and tried to initiate variables at the beginning of the function. Additionally, I believe that my code more closely followed the direction of Prof. Alcon. While it wasn't a hard-fast rule, it made the code much cleaner and easier to follow. Alex's code also used the vector pop_back and push_back methods to change the size of the returned vector. I believe this was slightly less efficient than iterating through a for loop and only assigning the values to the vector as needed.

I feel that Alex put much more thought in implementing his findMode function. With his use of a single for loop and some well-placed if statements, he was able to reduce the lines of code needed to implement the function. If he had provided more descriptive comments

throughout the program, I believe his function was the most creative of the group. I also think he made better use of the vector coding options.

Conclusion

The main thing I took away from this exercise was the uniqueness of code and the diversity of thought when approaching a coding problem. I also learned the importance of effective comments throughout code and why it is important to think of others when structuring a function. I learned that reading code can often be as important as writing code. I have often thought that programming is just one person sitting down at a keyboard and hammering out a function or widget. However, without the input of others, that single person will miss a failure in the logic or a possible improvement. This assignment helped me realize the importance of feedback and allowing others to identify improvements where I was previously blinded. Going forward, I will use the input from fellow programmers to identify holes or improvements in my code. Of course, that starts with an improvement in the readability and annotations in my widgets.