Module8，Assignment 7(comparisons)

WoohyukYang

yangwo@oregonstate.edu

For each pairwise comparison, where do you think your program is better? Why? Where do you think the other program is better? Why? (be detailed)

Firstly comparing my code with Salwa Mohamed's one, his work searches for mode correctly and can get the numbers even if every elements in the array appears only once, which means his program is not only able to search for the mode but also can get 1, 2, 3, and 4 if 1, 2, 3, and 4 appears only one time respectively. His program is better than mine when it comes to explanation. He attached appropriate comments to explain his code. It was easy to understand his logic.

Stephen Moran's program worked well like Salwa's program. But he used nested for loop, which means it takes more time to execute than my code and Salwa's code(Salwa and I did not use nested for loops, but we could still get the right answers for every desired case.) He also commented on his code in detail which is better than my program. He commented out some "cout" statements after testing his code. The code looked quite messy at first but it was helpful to test the code thoroughly in the end.

David Rider's code did not work correctly. It compiled but could only get a single number, which means in case 1, 2, 3, and 4 appears, the code could not get 4 numbers as answers. It still worked for a normal case like searching for a single mode. On the other hand, David's code was clean and clear to understand. He did not forget to comment on his logic also.

What have you learned from looking at other people's code, and how can you apply it in future assignments? (be detailed)

I could learn a lot from looking through Salwa Mohamed's code, he minimized the numbers of variables when writing the code and it made the code way simpler and easier to understand. I used extra variables just to get simple values which could be still calculated with usage of formerly set variables. When working with other people is required, minimalizing to assign useless variables could be helpful for the others to understand what I write. Other group members did not forget about commenting on their work in detail in regards to their logic and thoughts.

Spencer commented out the test cases on his code so I could see his code including parts which were commented out. Being provided with those proper test cases was really helpful for me to work on his code. I could see the flow of the program easily. I did not have to write down the values to track down the results of each loop. As a writer, guiding a proper way to test my code will be necessary. Then it must be easier to debug a program if it is equipped with test lines already.

David was really good at commenting on his overall logic before starting the code. I was guided to the right way at first so I could not be lost while analyzing the code. Like scholars who write down abstractions on the first page of dissertations, programmers need to write down their intentions too. I can definitely apply this good habit in future assignments.

Reading other group members' codes, I could find their bad habits and could not understand the purpose of each line from time to time. I could know why I need to make other people able to understand what I would write too. I recognized the importance of assigning appropriately desired words to variables. It was a good experience because I could see lots of other people's good coding styles.