

Which program did your group decide on?

Our group has chosen Shannon's implementation

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

Shannon's code was the easiest to read. It featured two stand alone for loops: one to find the highest frequency from the array passed in and another to compare each element count to the highest frequency. It also most closely followed the professor's example layout. Shannon wisely elected to sort the array at the start of the function, which made it easy to determine which element was a mode.

There were also some small things in the other implementations that were not featured in Shannon's that for the purpose of this class pushed us toward Shannon's:

- Stephen's code was a little more innovative in the fact that it used helper functions, but at least one of these were not necessary and could have been written in the `findMode` function.
- Sylvan's code used a series of nested for loops which can cause some performance issues, especially when the larger arrays are passed in his function.
- Jonathan's code used the least amount of loops, which is efficient, however; it used the `continue` keyword which is something we were warned to try to avoid. His code sorted his vector at the end. This caused an extra step of needing to verify the vector elements. If a new mode was found then the vector was cleared and a new element pushed in.

Shannon overall took the direct route and structured her code without any extras until the end. She sorted the array, tested for the frequency, and tested for the mode while pushing the mode into a vector. The last step was a final sort of the vector (which was not needed). The sorted array at the beginning made it so that the arrays mode could be verified naturally. There was no need for the extra step to verify the elements in the vector as we have seen in some of the other codes.

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

Shannon's implementation could use more comments, especially to describe the use of function parameters and variables. Sylvan's code is a good example of this. He used very descriptive comments that explained the precise function of the section of code. An example of this is:

```
// Second iteration through the array is to examine each element and  
if it occurs as much as  
// highest Occurrence, i.e. it is a mode, then add that value to the  
vector.”
```

Shannon’s design sorts both the input array and return vector, although the last sort is unnecessary as the modes are already inserted into the vector in ascending order.

The implementation could be made more readable by using some “helper functions” to break up the `findMode` function into smaller functions. Stephen’s code is a great example of this. By separating the code into functions, the work is split up so that a function is not performing too many tasks at one location. This separation can aid in easier code debugging or hand tracing.