

## CMPE322-HW2 REPORT

I created global variable for each value in file1. Then I wrote the functions needed to find those values. I wrote the mode finding function so that if there is more than one mode, it will find the smallest one. I divided it into  $N-1$  as desired in the deviation finding function. In the interquartile finding function, I evaluated 4 different situations. I determined different upper and lower values according to these situations.

I sorted the vector before calling the functions in the main, since the vector must be ordered in the median find and interquartile find functions. After that I started the time and called the functions because we will compare the time of the operations, so I started the time after sorting in file2 and file3 also. I set the precision to 5 for the required values. First, I found the time between functions in nanoseconds, then I divided it by  $10^9$  and printed it to the file in seconds.

I created 10 functions in file2 and run each of these functions in a separate thread. Since thread execute functions in parallel, it is faster than threadless execution. In  $10^6$  inputs, it gives results in approximately 0.06 seconds in file1 and in approximately 0.025 seconds in file2 with 10 threads. File3 gives about the same result as file2 but I created 5 threads in file3. There may be several reasons for this, according to what I researched on the internet, using more threads than the number of cores of the computer can cause overhead, so 10 threads may not be different from 5 threads because my computer has 8 cores. In addition, the operations performed on the functions are not very complex operations, so there may not be much difference. For example, this difference would be more noticeable if we had done an operation with  $O(n^2)$  complexity.