Design Document

**Introduction**

**Sorting algorithms will sort unsorted data into a numerically ascending order. This program runs a variety of different sorting algorithms on a list of numbers that are in a text file that is read and outputs to the console the result of each sorting algorithm. The program will output the result for the total time required to sort as well as the total number of operations required to sort the data.**

**Data Structures**

**The overall program relies on two data structures, an array of integers called numbers and an array of integers called reset. The reset array is used for reading in the information stored on the text document that stores the initial list of numbers and is used to initialize the numbers array every time a new sorting algorithm is called. These arrays are initialized to hold a constant number of integers that is determined by the integer e. There are a couple of other arrays and integers that are created locally within functions and in the main program that are used as needed.**

**A variable that is declared as the integer track is used as an incrementor for each sorting algorithm which is reset every time a new sorting algorithm is called.**

**Functions**

**This program uses 20 functions. These functions are broken down into blocks that are required for each sorting algorithm. Each of these 20 functions are required for one of the 10 sorting algorithms being tested. 10 of these functions are the parent function to the applicable sorting algorithm that will contain the main body of that algorithm. Most of the remaining 10 functions are called either recursively or as needed in a loop in the parent function depending on what is needed to perform the sort. These functions will run until the array is sorted then return that array back to the main program.**

**The Main Program**

**The main program will open the file using constants that are declared before runtime and will close automatically if the file fails to open. It then reads in the integers and stores those to the array reset, followed by closing the input file stream.**

**A variable in the main program will store the users input that is used in determining what sorting algorithm to run. After the user inputs a selection, the variable will be used to call the applicable sorting algorithm’s parent function and output the applicable results. If the program runs into an out-of-scope input, the program will continually ask for an in-scope input if there have been no sorting algorithms called in the runtime. Otherwise, the program will automatically close if an out-of-scope input is received.**

**This process will continue until the close program input is received or an out-of-scope result is received.**