ECE368
Project 1 Milestone 1
Chul Woo Park

In this project, I am going to implement Shell Sort with using insertion sort algorithm and bubble sort algorithm. The main purpose of this project is to improve the efficiency of sorting algorithm. Therefore, I will generate the counter increment after each incident when comparisons and moves occur during the method. First of all, I am going to implement six functions in this project and general idea for these functions are as follows: For the Load_File function, my code will basically read the file until it hits the EOF, which is the end of the list that is to be sorted. In addition, just for the first iteration, it will consider the first line of the file as the number of elements in the list and store it in variable Size. Save_File should be really similar to Load_File, but it will write to the output file instead of reading it from the input file. The output file created with Save_File function will have the same format as the input file that was read with Load_File function. These will be my basic ideas on file operations. Next, the most important part of this project will be the Shell_Insertion_Sort and Improved_Bubble_Sort. These two methods will be quite similar that it uses some specific sequences for deciding the gap of the array. Based on that gap decided by specific sequence, it will divide the entire array into smaller sub-arrays. The point is to get the list gradually sorted as it progresses throughout the algorithm. As of now, I am considering “for loop” for the gap generation, but it might change later when I actually get to the coding. Inside that gap generation for loop, I will put a while loop that runs until the sorting method for such gap value is done. Such sorting method will be either insertion sort or bubble sort. For example, if there are 8 elements in the list, {0,1,2,3,4,5,6,7}, and the gap is 2, it should sort {0,2,4,6} and {1,3,5,7} respectively. In order to do that, I will nest two for loops. The outer for loop will iterate for the number of gap. Thus, it will basically select the sub-array. The inner for loop will iterate until it hits the end of the list. However, the important thing in this for loop is that the index should be incremented by gap. In this way, it will run throughout each element in the sub-array and get it sorted. Lastly, Seq1 and Seq2 will be created with a loop that updates the value inside. Since the sequence should be in an ascending order, I might have to use a recursion in order to write in the output file from the end of the iterations. These are my basic and brief ideas on my project and these are subject to change during writing the code and debugging.