Dalton Sprinkle

Professor Kopp

CS 222 02

Final Project

Program Description

To explain my program from start to finish I will start with the sorting functions(Shell, Selection sort).These two functions are easy to tell which is the most efficient when using small values, however as the text file increases we can find the shell sort becoming more efficient than the selection sort and therefore with the size of file we are working with, the shell is the most effective. The shell function works on the basis of breaking the initial file apart and creating a sublist by choosing all items that are ‘x’ items apart(x being a changing variable). Whereas selection sort chooses a value and makes one change for every pass through the list before placing in the proper location. The code used to put these functions into use is also similar as I am creating lists to organize the initial text file in the same way starting by separating the date, minimum, and maximum temperatures. With these separated and identified by index it was easier to then take the minimum and maximum temperatures and use the desired calculation to change them into fahrenheit. From that point for both shell and selection sort I appended them to another list called ‘condensed’ which linked the date, and modified temperatures together into another full list. I also created a list called FullList which can call the full unedited version of the files if needed in a later use. The seperate lists of date, maximum, and minimum temperatures are still there in case they are needed for other calculations at a further date or to look at them as a whole without the other data files. There is also a count that lists the number of comparisons found within the text file that is shown at the end allowing the user to recognize how many lines of data were read. The conversions for the temperature was done on a read line basis converting the given temperature one at a time and then re-storing the value onto the seperate lists as said earlier. This gave me the option to store the initial value as well in case it would happen to be needed again in the future.

Moving on to the user search I chose to use a Linear Search for the benefit of not needing to have a ordered list to work on. This does increase the time needed to run the program however having the option to work with a unordered list gives a greater benefit in my opinion. This searching method is an inefficient technique compared to a binary search and so the code is pretty simple. To start I create necessary counters and again create a list called ‘condensed’ which is a list combining all values in the text file just with the converted temperature values. Once this is done I prompt for user input asking for a date in the appropriate format. When the date is given a for loop accesses the ‘condensed’ list and reads it iterance by iterance looking for a matching value, increasing the variables count(to keep track of iterations) and index(to keep index to be used to print corresponding temperature values). Then a if statement to be used when the value is found to print out the date, maximum, and minimum temperatures, utilizing the index to pull directly from the list the index of the next two values that correspond with the date and adding a +1 to searches to average number of comparisons. Lastly it prompts for user input again to repeat the process until the user enters ‘Done’. The only thing left out of this search function is when the user enters a wrong date or mistypes a value as for some reason I could not get it to work with the way I wrote my code.

The final part I will write about is the main program which calls all of these functions into action. I chose to go back to creating a menu which can allow the user to easily access which part they wish to see without being bombarded by everything at one time. It is based off a number system asking the user to enter a number 1-4 which corresponds to various functions. As said before though the UserSearch does not have a working ‘end’ option and will therefore require program termination to exit Nor is the final part of this assignment complete.