

Programming Assignment #4

Sorting

Sorting is the process of taking a bunch of items, all of the same type, and putting them into some defined order. For example, you might sort your CD collection alphabetically by the name of the artist or group. Books in a library are sorted by topic, then author name, e.g. by using the Dewey decimal system, or the newer standard you see in for example our TAMUC library today. We can also talk of sorting items in a computer program. Often we have a collection of items, held in an array, that we want to put into some order. If the items are integers, we might sort them in increasing numerical order, from the smallest item in index 0 to the largest integer in the last index of the array. If the array contains names, we might sort them alphabetically.

In this assignment we will modify the bubble sort and selection sort algorithms of the book to work with an array of strings.

Objectives

Work with two different sorting algorithms, bubble sort and selection sort. Compare the algorithms and modify them to work with strings rather than integers.

Instructions

1. Download the Sort.cpp file from the class web site as well as the unsortednames.dat data file. The data file is actually the same set of names from Lab 4 on using the binary and sequential search, except that the names have been all jumbled up. The Sort.cpp file contains an input function that reads the names from the unsortednames.dat file and places them in an array of strings for you to use.
2. Using the code for the bubble sort from the book as an example, implment a bubbleSort() function that will sort an array of strings, rather than integers. The bubbleSort() function should take 2 parameters, an array of strings containing the names to be sorted, and an integer which indicates the number of names in the list. The bubbleSort() function does not return an explicit result, so it is a void function, its results are returned implicitly by returning the array of strings back but now in a sorted order. Test your bubbleSort() on a small array of strings that you defined and initialize

yourself, to make sure it is working. Then see if it will work with the large list read in from the `unsortednames.dat` file.

3. Likewise, implement the `selectionSort()` function using the code from the book as your starting point. The `selectionSort()` takes exactly the same 2 parameters as the `bubbleSort()` the array of strings, and an integer with the number of strings to be sorted. Test your `selectionSort()` in the same way, first with a small array of strings that you create, then with the large list of names from the data file.

Program 4 Finished

You have now completed Program 4. As with previous assignments, the final step you should perform when you complete an assignment is to upload your finished assignment to your eCollege account. Go ahead at this point and upload your `PlayBall.cpp` source file to eCollege. Once this is done you have successfully completed the programming assignment #4.