CS1 Spring 2019

Assignment 3: Linear Insertion Sort

We have talked about two sorting algorithms, Bubble Sort and Selection Sort. This one is slightly different and involves several concepts we have discussed. Here is the method:

1. You can use a statically-allocated one-dimensional array of doubles for this with length 100. You will never have more than 100 things to sort, but you may have fewer. Your program will have to keep track of how many elements you actually use.
2. Ask the user for the name of a file containing data. If it does not exist, the program should display an error, then ask for a new file name. Entering an asterisk (\*) as the first and only character on a line should terminate the program.
3. The way linear insertion works is this:
4. If the list is empty, the number goes in the first array element.
5. If the list is not empty, search to find the two elements where the top one is smaller than the one you have and the bottom one is larger (assuming an ascending-order sort.) This is your insertion point. Then move everything below and including the insertion point down in the array. Put the new element in.
6. Increment the count of elements in the array
7. Once the list is sorted, print it.
8. Go back to step 2 and ask for another file name.

For example, if the file contains 6, 3, 7, and 2 your array would look like this at each step:

|  |  |
| --- | --- |
| Step 1: Array is empty, so insert the 6. | 6 |
| Step 2: Move the 6 down, insert the 3 | 3  6 |
| Step 3: Put the 7 at the end; nothing moves | 3  6  7 |
| Step 4: Move the 3, 6, and 7 down, then insert the 2. | 2  3  6  7 |

Yes, this is terribly inefficient because of moving elements around, but it illustrates several important concepts, including sorting, functions with parameters and return values, file I/O, and array handling.

There are two test files on eLearning. Download these and test your program with them. You can assume that the files do not contain bad data. They contain only valid floating-point numbers. Make no other assumptions about the data.

**To hand in:**

Your **.CPP** file named <NetID>Asg3.cpp

|  |  |
| --- | --- |
| **Grading Criteria** | |
| Program behaves according to the specification, sorting correctly, requesting file names, and catching “file not found” errors. | 60% |
| Program is structured well, with functions to do most of the work rather than everything in the main loop. | 30% |
| Program documentation | 10% |

**Grading Rubric:**

Bad output: -30

Using a different algorithm from the one described above: -30

Missing internal comments: -5

Program does not use functions: -20

Program does input or output in the same function that does the sorting: -20

Program sorts only one file and stops: -10