CSCI/CMPE 1370

Homework Assignment #3: Arrays of numbers

The problem:
Write a C++ program that reads a list of numbers from a file into an array, then uses that array to find the average of all the numbers, the average of the positive and negative numbers (0 is neither positive nor negative!), and the largest number. The point is to be able to store data in an array and do things with it.

Submitting:
Once again, you will have the option to work in randomly assigned pairs. The ability to work effectively with others is a critical skill. The teamwork policy is described in the course syllabus. You will turn in **one** copy of the assignment as a team. Your submission must have the following comment at the top:

```
//****************************************
// First and last name of team member 1, First and Last Name of team member 2
// CSCI/CMPE 1370
// Homework Assignment #3
//****************************************
```

Part of this assignment is to come up with a set of tests that prove that your program works. **Include sample runs** of the program. These test runs should be copied into a separate text file and submitted along with your code file.

As before, there is a self-evaluation form to help you evaluate your work and show that you are able to do so. Copy the form below into a new text file, fill it out, and submit it with your code. The numbers in parentheses are the max points for each requirement.

When you are satisfied that your solution is complete, submit it ***through blackboard.*** *Your submission should include at least one .cpp file and one .h file containing your program, a separate .txt file with your self-evaluation, and a separate .txt file with your sample runs.*

Details:
Your program must contain at least the following four functions, **which must be in a separate header file** (not in with main).

1. `read_list`

    This function will take as input parameters an array of integers and a string filename. It will open that file and read in numbers, storing them in the array, stopping at the end of the file. Remember that there are tricky issues with extra whitespace at the end of the file. You can assume the file contains only integers. The function will return the number of numbers that were read in.

    Note: In versions of Visual C++ pre-2010, when using the `open` function with a string argument, you must call the `c_str()` function on the string variable, such as:

    ```
    my_input_stream.open( filename.c_str() )
    ```

2. `averages`

    This function will take as input parameters an array of integers and the length (the number of valid numbers) in that array. It will pass back 3 real number values: the average of those numbers, the average of the positive numbers and the average of the negative numbers. 0 is not considered positive or negative for this assignment.

3. `largest`

   This function will take as input parameters an array of integers and the length (the number of valid numbers) in that array. The function will return the largest number in the array.

4. `display`

   This function will take as input parameters all the values of interest from the first 3 functions. It will print to the screen a report that looks like:

   ```
   10 numbers read from file "data.txt".
   The average of all the numbers is: 17.3.
   The average of the positive numbers is: 23.5.
   The average of the negative numbers is: -5.1.
   The largest number is: 61.
   ```

Input:
You (and we) will want to test your program on multiple data files. You could hardcode the file name for each run, or you could stop and ask the user to type in the file name, but both require extra effort from the tester. Instead, you are going to automate running more than one test.

In main, you must:

1. Declare an array to hold some number of file names (and an integer indicating how many names are in the array)
2. Set the file names you want to test by putting them in the array (hardcoded, not from the user)
3. Loop over that array processing each file one after another

You must create at least two separate data files (e.g. data_a.txt and data_b.txt) with the following data in them:

Data Set A:
4 -30 0 7 42 -20 18 400 -123 -6

Data Set B:
2 17 -5 0 20 15 -16 -3 -2 14 -1 12 1 -5 -100 15 22 -5 68 -13

Include the output of your program on those two files, either as a comment or as a separate text file!

Self-Evaluation Form

The required functions:

read (75):
averages (100):
largest (75):
display (50):

Loop over multiple files (50):

Good coding practices:
Style (indenting & spacing, reasonable identifiers, file structure) (50):
Compiles and runs (50):
Sample test runs (25):

Self-evaluation (25):

Comments: