Homework 1

Arrays, Strings, Structures, Pointers, Functions and Files

100 Points

Projects:

26B_H_1A.c – Process an array of structures (detailed instructions are given as comments)

26B_H_1B.c – **Create and process an array of structures** – see program requirements on the next page.

Grading

Program 1A – 25Points

1. The printList() function – 10
2. Second sort – 10
3. Updated main() – 5

Program 1B – 70 Points

1. Read file names – 5
2. Read from file – 25
3. Print Data Manager – 20
4. Sort Array – 10
5. Write to file – 10

Self Assessment Report: – 5Points

Write a short report (**26B_H1Report.docx**), briefly explaining your code and containing an assessment of your implementation based on the above grading criteria.

Your assignment will be graded on correctness, structure, style, clarity, and documentation (see Homework 0):

- Write cohesive functions.
- Write a comment in the beginning of the program. Write a comment for each function definition. Write comments inside functions (as needed).
- Use proper indentation and spacing.
- Do not use global variables.
- Do not use the goto statement.
- Always check if opening an input file is a successful operation.
- Assume input files have valid data and consistent formatting (no validation needed).
- Always validate the user's input.

Project B: Create and Process an Array of Structures

Write a program that provides information about two-tear colleges in California. The program expects the name of an input file and an output file to be given by the user. If the user does not input any names, default file names should be used, such as **`in.txt`** and **`out.txt`**. The input files have lines which look like this:

```
3 Santa Barbara City College; 1909 30687
```

The first number represents the rank; it is followed by the school name, year founded, and approximate number of students.

Read the list of colleges into an array of structures. You may assume that the maximum size of a college name string is 64.

The program should use either the insertion sort algorithm or the selection sort algorithm to sort the array in ascending order by rank. The college that is ranked #1 will be displayed first, then #2, #3, and so on.

Display any k consecutive schools requested by the user. For instance, if the user enters 1 2, the program displays the first 2 schools in a readable format of your choice. If the user enters 1 5, the program displays the first five schools, if the user enters 2 7, display 7 schools beginning with the one at index 1 (2 - 1), and so on. Repeat this process until the user enters 0 0. Reject invalid input (such as -1 5, or 1 200, etc.)

Finally write the sorted array to the output file, using the same format as in the input file's format.

Run the program once and save the screen output at the end of the source file as a comment.

What to upload? Compress source files, input and output files if any, the self assessment report, and nothing else. Upload the compressed file: 26B_LastName_FirstName_H1.zip

Input: Use the given input file or create your own input file using the data shown below.

`in.txt`

```
3 Santa Barbara City College; 1909 30687
5 Pasadena City College; 1924 22000
14 West Hills College – Coalinga; 1932 4000
7 Napa Valley College; 1942 8996
2 Orange Coast College; 1947 25000
15 Palo Verde College; 1947 3898
4 Diablo Valley College; 1949 24781
6 Foothill College; 1957 18362
12 College of the Siskiyous; 1957 2473
10 Cuesta College; 1963 9571
8 Ohlone College; 1965 18000
1 De Anza College; 1967 24781
11 Feather River Community College District; 1968 1635
13 Lake Tahoe Community College; 1975 3000
9 Irvine Valley College; 1985 14384
```