CS1050 – Lab 10

Spring 2020

Concepts to Practice

- Structures
- Arrays of structures
- Sorting
- Formatted I/O

Submission Information

Submit this assignment by following the instructions given by your TA. SUBMIT ONLY the .c file (no a.out or executable file is required). All of the lab assignments must be submitted before the end of the lab using the lab code given by the TA.

Use the following submit command:

```
mucs submit <class> <assignment_type> <filename>
```

For example:

```
mucs submit 1050 lab x-lab10.c
```

Filenames must be: *sectionletter-lab10.c* (include your respective lab section, e.g., a-lab10.c).

Description

For the lab assignment, your program will be given a command-line argument specifying the name of the file to read in, an optional second argument specifying a field by which to sort (discussed below), and an optional third argument specifying ascending or descending order. The file is the same as your prelab. Each line in the file represents a movie. The data in each movie is:

- Title
- Gross Revenue (adjusted for inflation to 2019 dollars)
- Year

You are to read in the movies, and output them depending on the sorting parameter given on the command-line. If no sorting parameter is given, you should sort them alphabetically by title. If the optional order argument is specified, you will use it to determine whether to sort either ascending or descending. If no optional order argument is specified, you should assume ascending order. If the sorting field (2nd argument) is not specified, there can be no order argument (3rd argument).

Possible values of arguments:

- Filename – 1st argument – required
- Sort field – 2nd argument – optional. Must be "Title", "Gross", or "Year" (otherwise this is an error)
- Sort order – 3rd argument – optional (and only if 2nd argument specified). Must be "Ascending" or "Descending"

You can get the data file from tc.rnet.missouri.edu in the /group/cs1050/data directory. You can copy it to your local directory via:

```
cp /group/cs1050/data/prelab10.dat .
```

BONUS

For bonus points, determine the number of movies in the file and then dynamically allocate your array of structures rather than hard coding the number of movies up front. This would allow your program to work with larger files of data.

Sample Output

Example 1

```
jimr@JimRArea51:~/CS1050/SP2020/lab10$ ./a.out
./a.out filename [sort_field [sort_order]]
         - sort_field must be one of "Title", "Gross", or "Year" if specified.
         - sort_order must be one of "Ascending" or "Descending" if specified.
```

Example 2

```
jimr@JimRArea51:~/CS1050/SP2020/lab10$ ./a.out asdfasdf
*** Error: Cannot open file asdfasdf
```

Example 3

```
jimr@JimRArea51:~/CS1050/SP2020/lab10$ ./a.out prelab10.dat
Sorted movies:
                         Title                Gross                 Year
                        Avatar           3263000000                 2009
             Avengers:_Endgame           2798000000                 2019
                Doctor_Zhivago           2238000000                 1965
    E.T._the_Extra-Terrestrial           2493000000                 1982
            Gone_with_the_Wind           3713000000                 1939
                     Star_Wars           3049000000                 1977
  Star_Wars:_The_Force_Awakens           2206000000                 2015
            The_Sound_of_Music           2554000000                 1965
          The_Ten_Commandments           2361000000                 1956
                       Titanic           3087000000                 1997
```

Example 4

```
jimr@JimRArea51:~/CS1050/SP2020/lab10$ ./a.out prelab10.dat asdf
./a.out filename [sort_field [sort_order]]
         - sort_field must be one of "Title", "Gross", or "Year" if specified.
         - sort_order must be one of "Ascending" or "Descending" if specified.
```

Example 5

```
jimr@JimRArea51:~/CS1050/SP2020/lab10$ ./a.out prelab10.dat Gross
Sorted movies:
                         Title                Gross                 Year
  Star_Wars:_The_Force_Awakens           2206000000                 2015
                Doctor_Zhivago           2238000000                 1965
          The_Ten_Commandments           2361000000                 1956
    E.T._the_Extra-Terrestrial           2493000000                 1982
            The_Sound_of_Music           2554000000                 1965
             Avengers:_Endgame           2798000000                 2019
                     Star_Wars           3049000000                 1977
                       Titanic           3087000000                 1997
                        Avatar           3263000000                 2009
            Gone_with_the_Wind           3713000000                 1939
```

Example 6

```
jimr@JimRArea51:~/CS1050/SP2020/lab10$ ./a.out prelab10.dat Gross asdf
./a.out filename [sort_field [sort_order]]
         - sort_field must be one of "Title", "Gross", or "Year" if specified.
         - sort_order must be one of "Ascending" or "Descending" if specified.
```

Example 7

```
jimr@JimRArea51:~/CS1050/SP2020/lab10$ ./a.out prelab10.dat Gross Descending
Sorted movies:
                         Title                 Gross                 Year
            Gone_with_the_Wind            3713000000                 1939
                        Avatar            3263000000                 2009
                       Titanic            3087000000                 1997
                     Star_Wars            3049000000                 1977
             Avengers:_Endgame            2798000000                 2019
            The_Sound_of_Music            2554000000                 1965
    E.T._the_Extra-Terrestrial            2493000000                 1982
          The_Ten_Commandments            2361000000                 1956
                Doctor_Zhivago            2238000000                 1965
  Star_Wars:_The_Force_Awakens            2206000000                 2015
```

Example 8

```
jimr@JimRArea51:~/CS1050/SP2020/lab10$ ./a.out prelab10.dat Year Ascending
Sorted movies:
                         Title                 Gross                 Year
            Gone_with_the_Wind            3713000000                 1939
          The_Ten_Commandments            2361000000                 1956
            The_Sound_of_Music            2554000000                 1965
                Doctor_Zhivago            2238000000                 1965
                     Star_Wars            3049000000                 1977
    E.T._the_Extra-Terrestrial            2493000000                 1982
                       Titanic            3087000000                 1997
                        Avatar            3263000000                 2009
  Star_Wars:_The_Force_Awakens            2206000000                 2015
             Avengers:_Endgame            2798000000                 2019
```

Guidelines for Grading Lab 10
40 Points Possible (+5 bonus points)

General

If your program does not compile or produce any input/output (I/O) because most of the source code is commented out then your lab will receive a grade of ZERO POINTS. Further, if your program does not actually follow the specifications, but merely prints out lines that make it appear to follow the specifications, you will receive a grade of ZERO POINTS. For partial credit your C program must not only compile but also produce some valid I/O that meets the lab specifications.

You program is expected to have a comment header at the top that includes your name, pawprint, the course you are taking, and the lab that you are solved (e.g., "Lab 10"). Your code should be nicely indented. **You will lose up to 10 points if you do not meet these basic requirements**.

5 points: Your code error-checks that the specified file can be opened and read.
5 points: Your code properly closes opened files in all cases.
5 points: Your code error-checks that the optional sorting field argument is one of the legal values.
5 points: Your code error-checks that the optional sorting order argument is one of the legal values.
5 points: Your code sorts by the correct field.
5 points: Your code sorts in the correct direction (ascending or descending)
10 points: Your output closely matches the example output.

BONUS (5 points)

5 points: You dynamically allocate an array of structures, based on the number of records in the file.