CSCI 1470 **Fall 2024** **Mr. Gustavo Dietrich**

Lab 20: Array of structs **Due Date:** 11/29/24

In this lab you will practice working with structs, and with searching and sorting algorithms.

Example Program

This example reads the grades for each student into an array of structs that hold student grades. It also uses sorting and searching algorithms to find a specific student's grades.

The example program reads the **classGrades.txt** file. Each line of the file contains the student's name, id, and three exam grades. The **classGrades.cpp** program provided illustrates the skills you are learning in this lab. Open it in your IDE, read and understand the code and finally run the program to see how it works.

Your Program

Write a program that reads information about movies from a file named **movies.txt**. Each line of this file contains the name of a movie, the year the movie was released, and the revenue for that movie. Your program reads this file and stores the movie data into a list of movies. It then sorts the movies by title and prints the sorted list. Finally, your program asks the user to input a title and the program searches the list of movies and prints the information about that movie.
To represent the movie data you have to create a struct named **Movie** with three members: title, yearReleased, and revenue.
To handle the list of movies you have to create an array of **Movie** structs named **topMovies** capable of storing up to 20 movies.

Your solution must be implemented using the following functions:

storeMoviesArray(ifstream inFile, Movies topMovies[], const int SIZE)
// This function receives the input file, the movies array, and the size of the
// array.
// It reads from the file the movie data and stores it in the array.
// Once all the data has been read in, it returns the array with the information
// of the movies.

sortMoviesTitle(Movie topMovies[], const int SIZE)
// This function receives the movies array and the size of the array and sorts
// the array by title. It returns the sorted array.

printMoviesArray(Movie topMovies[], const int SIZE)
// This function receives the movies array and the size of the array and prints
// the list of movies.

findMovieTitle(Movie topMovies[],const int SIZE, string title)

// This function receives the movies array, the size of the array, and the title
// of the movie to be searched for.
// It returns the index of the array where the title was found. If the title was
// not found, it returns -1. **It must use binary search to find the movie.**

main()
// Takes care of the file (opens and closes it).
// Calls the functions and asks whether to continue.

Open **lab20.cpp** in your IDE and implement the algorithm provided in the source code as comments according to the above specifications.

Note: all the functions must be defined in this file **below** main()

IMPORTANT:

- You must define whether the provided functions are a void or a value-returning function. You CANNOT add or remove functions.
- You must define whether the provided parameters are a value parameter or a reference parameter. You CANNOT add or remove parameters.
- You must use the identifiers I provide for the parameters and functions.
- Assume the input file will always have exactly 20 movies.
- Make sure you don't make unnecessary calls to the functions (if only one call is needed then call it just once).

Review the examples discussed in class and the textbook to get an idea of what you need to do. **Include comments in the source code to make it as readable as possible.**

IMPORTANT:

For your reference on how the program should interact with the user I am providing you sample runs of my solution. Observe them carefully and make sure your program behaves similarly to mine (pay attention to the output generated).

Your program must be well commented, use meaningful identifiers, **use efficient structures**, and use indentation as shown in the textbook.

Don't forget to include at the top of the program the comments shown below with your information (name, class and section number, etc.)

```
//////////////////////////////////////////////////////////////////////
// Name: <Put your name here>
// Date: <Today's date>
// Class: <Your class number and section number, like: CSCI 1470.02>
// Semester: <This semester, like: Fall 2012>
// CSCI 1470 Instructor: <Your lecture instructor's name>
//
// Program Description: Enter here your description of what the program does
//
/////////////////////////////////////////////////////////////////////
```

When done, submit your solution through Blackboard using the "Assignments" tool. Do Not email it.

Paste the link to your final solution along with your source code in the textbox opened when you click on Create Submission before you click on Submit.

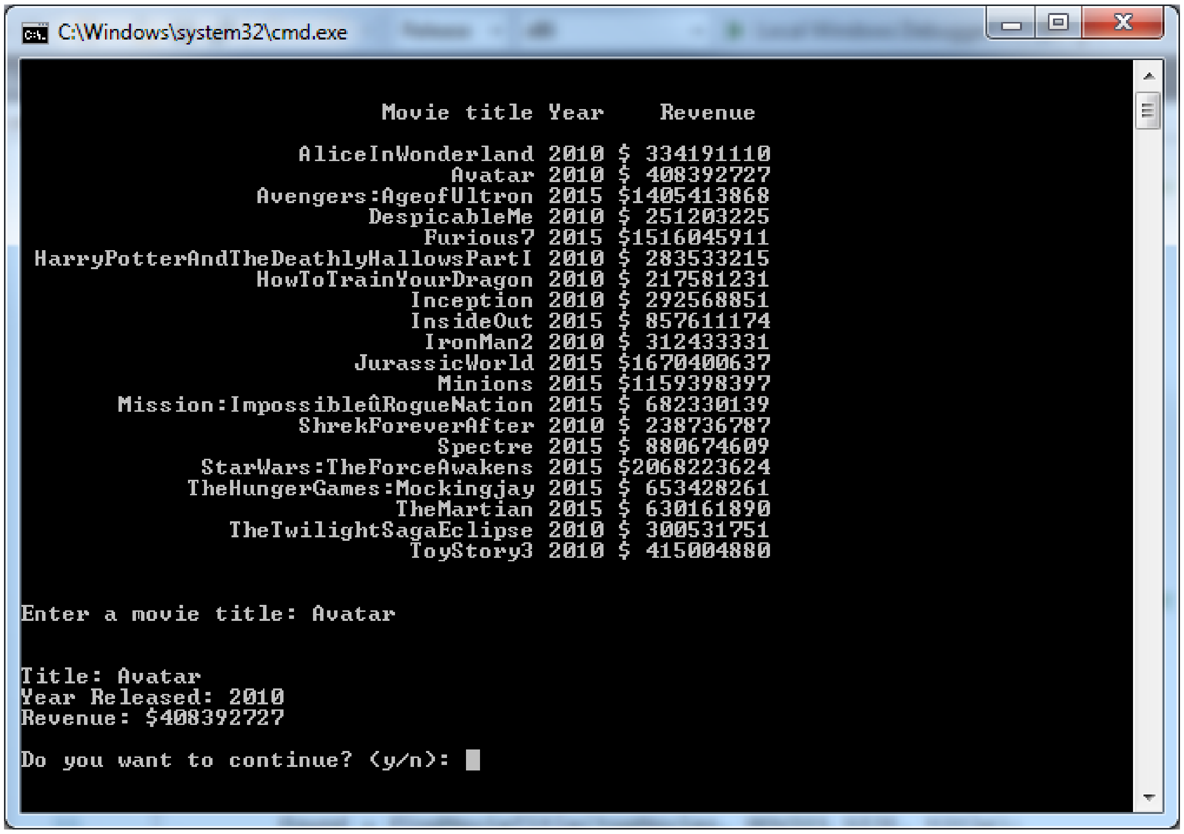
Grading criteria

You start with 100 points and then lose points as you don't do something that is required.

-5: Minor mistakes
missing/too few comments
didn't follow the specified format for input/output
didn't accept answer in upper or lower case (use of toupper(), tolower(), or || are accepted)
didn't use value and reference parameters or function value data type correctly (each function)
-10: Moderate mistakes
wrong implementation of the struct for the movies data
wrong implementation of a required function (each)
incorrect function name (each function)
incorrect function call (each function)
didn't use loops where appropriate
unnecessary calls to a function
-20: Major mistakes
didn't implement the required functions (each)
program does not implement the provided algorithm
Incorrect/missing cpp file or incorrect/missing link to your solution
-40: Program doesn't run correctly
-50: Program doesn't compile
-100: The code submitted is not your creation (you got it from a web site or another person)
-10: Late

Important: more points may be lost for other reasons not specified here.

Sample runs



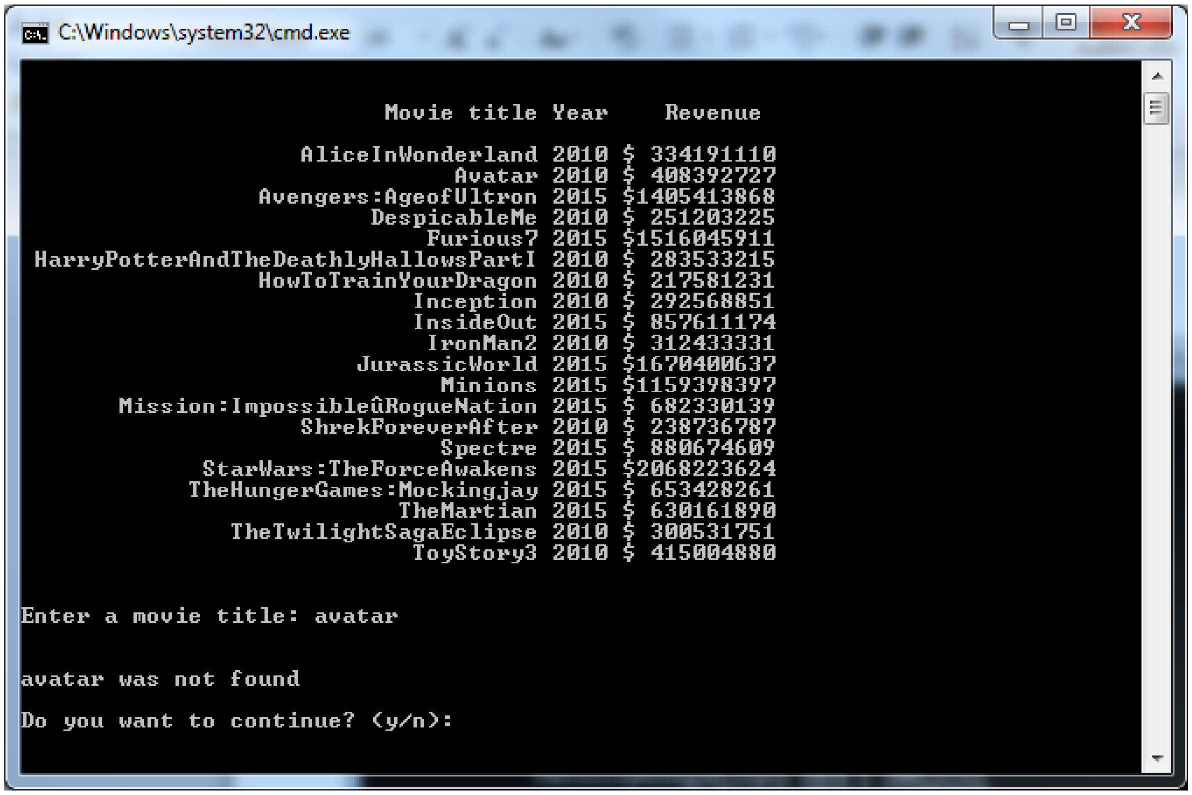
```
                          Movie title Year     Revenue

                    AliceInWonderland 2010 $ 334191110
                               Avatar 2010 $ 408392727
                 Avengers:AgeofUltron 2015 $1405413868
                         DespicableMe 2010 $ 251203225
                             Furious7 2015 $1516045911
 HarryPotterAndTheDeathlyHallowsPartI 2010 $ 283533215
                 HowToTrainYourDragon 2010 $ 217581231
                            Inception 2010 $ 292568851
                            InsideOut 2015 $ 857611174
                             IronMan2 2010 $ 312433331
                        JurassicWorld 2015 $1670400637
                              Minions 2015 $1159398397
       Mission:ImpossibleûRogueNation 2015 $ 682330139
                    ShrekForeverAfter 2010 $ 238736787
                              Spectre 2015 $ 880674609
             StarWars:TheForceAwakens 2015 $2068223624
            TheHungerGames:Mockingjay 2015 $ 653428261
                           TheMartian 2015 $ 630161890
               TheTwilightSagaEclipse 2010 $ 300531751
                            ToyStory3 2010 $ 415004880


Enter a movie title: Avatar


Title: Avatar
Year Released: 2010
Revenue: $408392727

Do you want to continue? (y/n):
```



```
                          Movie title Year     Revenue

                    AliceInWonderland 2010 $ 334191110
                               Avatar 2010 $ 408392727
                 Avengers:AgeofUltron 2015 $1405413868
                         DespicableMe 2010 $ 251203225
                             Furious7 2015 $1516045911
 HarryPotterAndTheDeathlyHallowsPartI 2010 $ 283533215
                 HowToTrainYourDragon 2010 $ 217581231
                            Inception 2010 $ 292568851
                            InsideOut 2015 $ 857611174
                             IronMan2 2010 $ 312433331
                        JurassicWorld 2015 $1670400637
                              Minions 2015 $1159398397
       Mission:ImpossibleûRogueNation 2015 $ 682330139
                    ShrekForeverAfter 2010 $ 238736787
                              Spectre 2015 $ 880674609
             StarWars:TheForceAwakens 2015 $2068223624
            TheHungerGames:Mockingjay 2015 $ 653428261
                           TheMartian 2015 $ 630161890
               TheTwilightSagaEclipse 2010 $ 300531751
                            ToyStory3 2010 $ 415004880


Enter a movie title: avatar


avatar was not found

Do you want to continue? (y/n):
```