CS201 – Fall 2020-2021 - Sabancı University
Homework #5 – Netflix TV Series Ratings
Due December 23rd, Wednesday, 23:55 (Sharp Deadline)

Brief Description

In this homework, you will write a program that produces an output based on data read from two different files. Your program will read Netflix TV Series names and actors from one file, and TV Series ratings from several critics from another file. By processing these two different data, the program will allow the user to select an option within a menu structure and provide a different information on each option on the screen. The details of the input and output file formats, and the rules of processing will be explained below.

Your take-home exams will be automatically graded using GradeChecker, so it is **very important to satisfy the exact same output given in the sample runs.** You can utilize GradeChecker (http://learnt.sabanciuniv.edu/GradeChecker/) to check whether your implementation is working in the expected way. To be able to use GradeChecker, you should upload all of your files used in the take-home exam (**cpp, header and txt files for this take-home exam**). Additionally, you should submit all of your source (**cpp and header**) files to SUCourse. **Just a reminder, you will see a character ¶ which refers to a newline in your expected output.**

The name of your main source (cpp) file should be in the expected format: "SUCourseUsername_THEnumber.cpp" (all lowercase letters). Please check the submission procedures of the take-home exam, which are listed at the end of this document.

Name and Structure of Database (Input) Files

The data is stored in two different text files. One lists Actor-TV Series pair records and the other one lists scores obtained from critics for a particular TV series. These two files will be input to your program (i.e. your program will read some data from both files). They are generated outside of the program using Notepad (we provide these files on google drive).

At the beginning of your program, the user will be asked to enter an input file name for the Actor-TV Series records. If the user enters a name for a file that does not exist, your program should display an appropriate message and ask for the name of a new input file name until the file is successfully opened. After the first input file is opened successfully, your program will ask for another input file name for the TV Series rating records file. The same file opening check will be performed for this file too. If the user enters a name for a file that does not exist, your program should

display an appropriate message and ask for the name of a new input file name until the file is successfully opened.

Ratings File

The ratings input file contains rating information for each TV series. For a single TV series, there are two columns of information on the same line. Each line corresponds to the record of a TV series. The record structure for each TV series is described as follows:

Rating	*TV_Series*

- Record line starts with the rating points of the TV series.
  - *Rating* is a single **real** number between 0 and 10 used to define the rating of the TV series by a critic such as 7.6, 10, 3.0 etc.

- Second and last data item of the record is the name of the TV series: *TV_Series*.
  - *TV_Series* may consist of more than one word. You may assume that the *TV_Series* name has at least one word in it, but there is no upper limit on the number of words. In the context of this homework, a word is defined as a sequence of non-space characters. A word may contain letters, digits, or any punctuation marks and symbols. Words that make up the *TV_Series* name may be separated by one or multiple white spaces. You should handle it.

The above-mentioned record structure, which contains 2 different data items in one line, is just for one *TV series*. The input file contains information about several *TV series* as well as multiple ratings for the same *TV series*. Therefore, there are several such records in the input file. You may assume there is a record in each line of input file, no line is empty.

Actors File

The Actors database file contains Actor-TV_series pairs where that actor is acting in the TV series. For a single actor record, there are 2 pieces of information on the same line. The record structure for each Actor-TV_series pair is described as follows:

Actor_Name;	*TV_Series*

- *Actor_Name* is the name of the actor and contains one or more words. For example, Millie Bobby Brown.
- *TV_Series* is the name of the TV series and contains one or more words. For example, Game of Thrones

We assume that an Actor-TV_series pair does not appear more than once in the Actors database file. For example, if there is a pair such as ActorName1-Series11, then there will not be another pair such as ActorName1-Series11.

There might be any number of white spaces (i.e. blanks) between and after each data item in both files. You should handle it.
The matching of TV series names should be case-insensitive. That is, a TV series name such as "FriEnds" in one file and "friends" in another file **SHOULD be considered the same**. Hence when searching for "FriEnds" found from the ratings file, you should match the word "friends" in the actors file, even if it is written all in uppercase or lowercase etc.

There is no specific order of records in both input files. That means you cannot assume that the records of the input files are ordered according to TV_*Series*, *Actor* or any other data attribute.

The structure of the input files and the associated rules and assumptions explained above are fixed: you cannot change them or make any other assumptions. No format check for the contents of the input files is needed: you may assume that all data are in the correct format. You may examine the sample input files (actors-series.txt, ratings.txt) provided in the google drive for this homework (THE5).

Processing

You **MUST** use a struct data structure to encapsulate the data of one TV series, such as TV series name, actor name, rating points, etc. It is up to you to decide which fields are to be included in this struct.

You **MUST** use a vector to store several TV series' data. Of course, the element type of this vector will be the previously defined TV series struct. Here please remark that you CANNOT make any assumptions about the number of TV series. Therefore, your program should work for any number of TV series. Moreover, your program should work for any TV series names. In the sample, we used real TV series names, but we can use other names for grading purposes.

Flow of the program may be as follows:

- Your program should first ask for the file name information from the user for input files one by one. For each input file name, you should try to open the file; if it fails your program should ask for another name until a valid file name is entered.
- Then, your program processes the input files. The processing method could be word by word or line by line.
    - As mentioned above, there are several rating points for a TV series in the input file. The processing for each unique TV series with a rating point in a line is as follows:

    - If this particular TV series is not in the vector (i.e. if this is the first appearance of the TV series with a rating), then create a new TV series struct for this TV series and add it to the vector.
    - If this particular TV series is already in your vector, update this existing struct. That is, if this TV series has occurred before in the ratings calculations, then there should be a struct in the vector for it. If this is the case, you should only update the rating information about the TV series without adding a struct for this TV series to the vector again.
    - Update the total rating point of this TV series by adding the points gained from this rating to its previous rating points.
    - You would also need to keep a count of ratings for each TV series in the TV series struct, as you will use the average rating of a TV series at the end.
- After your program finishes reading and processing all of the rating results from the input file and storing the necessary data in the vector, the program should provide menu structure and provide different information on each option on the screen. Details are described in the Output section.

Output

After your program finishes reading and storing data, it should then continuously show a menu as an action list for the user. The user will select one of these actions from the menu by entering its respective number between [1-5].

1. Print the series
2. Search the series according to the rating
3. Add the series to the favorite list
4. Print the favorite list
5. Exit

Your program will then check the validity of the entered value. If it's between [1-5] it will do the desired operation. Else (if it is out of range or not an integer at all) it should print an error message (`"This is not a valid option!"`) and print the menu again for a new input. Below, you can find in detail how the program will take action for each option.

Option 1 : Print the series

Your program should print the sorted ranking table into a console in a descending manner according to the average rating points of the TV series.

The format of the table is described below, and **you must follow this format precisely**.
The first line of the output file is the header line of the table. This line is fixed and given below:

RANK,NAME,ACTOR,POINTS

After this header line, the point table is displayed in rank order (from top to bottom). Rank values start from 1. You should display separate TV series data in separate lines.

For each TV series, four pieces of information must be displayed in the output when this option is chosen; these are the *Rank of the TV series in the point table*, the *Name of the TV series*, the *Name of the actor, the Average Rating Point that the TV series gained*. **These 4 pieces of information must be displayed in this order and there must be commas in between them.**

The TV series name always **MUST** be output in all **UPPER-CASE** no matter how it is spelled in either file. For example, the TV series name "dark" in TV series file and "DARK" in actors file is output as "DARK" in the console anytime.
Note: If multiple TV series have the same rating point, then they should be sorted by their TV series name; series with alphabetically smaller names (in string comparison terms) must be at a higher position in the table.

Option 2 : Search series according to the rating

When the user selects this option, the user will be asked to enter a minimum rating (double type) first.

```
Please enter the minimum rating:
```

Later, a table in the same table format as in **option 1** will be printed on the console, but this time only a TV series with a rating higher than or equal to the minimum rating given by the user will be printed.

Here, the user can enter a number with a maximum of 10 and a minimum of 0, if user enters an input in a format other than a double or a number out of limits (0-10), you should print the following message:

```
This is not a valid rating!
```

Option 3 : Add series to the favorite list

When this option is selected, the program will ask the user to enter a TV series name first as follows:

```
Please enter the name of series:
```

Afterwards, the TV series name that the user named as input will be added to the favorite list. Two points need to be noted here. First of all, if the data we have does not contain an TV series with that name, the following message should be written to the console:

```
There is no such TV series!
```

The second point is that if the user tries to re-add a previously added TV series, the following error message should be written to the console. Example: Let's assume

the user wants to add to their favorite list named dark, but it has been added before.

```
Your favorite list already have DARK
```

Option 4 : Print the favorite list

When this option is selected, the program should print the TV series user added to the favorite list in the table format depicted in **option 1**.

If the user wants to print the list when the list is empty, the following message should be printed:

```
Your favorite list is currently empty!
```

Option 5 : Exit

When this option is selected, the program should be terminated.

Hints & Important Points

- You may modify and use one of the sorting algorithms provided by the book and/or discussed in the lecture. If you want to develop your own sorting algorithm, of course you may, but this will be more difficult.
- You should sort the vector by rating points; TV series with higher points must be at higher positions in the table. If multiple TV series have the same point, then they should be sorted by their name; series with alphabetically smaller names (in string comparison terms) must be at a higher position in the table.
- You can keep another vector for the favorite list. Don't forget to close the files you open.
- Note that the menu should always appear on the console until the user selects the "Exit" Option.
- Review sample runs for the rest of the format

Sample Input and Output Files

In the Google drive of this homework, we provide sample input files: `actors-series.txt` and `ratings.txt`. You may examine them to understand the homework and input file format in detail.

Sample Runs

We do not provide a greetings part in the following sample runs, but you are expected to display an informative greeting. The *italic* and **bold** phrases are inputs taken from the user.

Sample Run 1

```
Welcome to Netflix TV Series Ratings Calculator!
Please enter a filename for Series-Actor file: actors.txt
Can not find the specified file.
Please enter a filename for Series-Actor file: series_actors.txt
Please enter a filename for Rating file: ratings
Can not find the specified file.
Please enter a filename for Rating file: ratings.txt
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add the series to the favorite list
4. Print the favorite list
5. Exit
---
Enter your choice: 1
---
RANK,NAME,ACTOR,POINTS
1,DARK,LOUIS HOFMANN,9.8
2,THE WITCHER,HENRY CAVILL,9.75
3,SEINFELD,JERRY SEINFELD,9.55
4,THE WALKING DEAD,NORMAN REEDUS,9.55
5,LUCIFER,TOM ELLIS,8.9
6,BREAKING BAD,BRYAN CRANSTON,8.7
7,MODERN FAMILY,SOFIA VERGARA,8.6
8,COMMUNITY,GILLIAN JACOBS,8.5
9,THE PUNISHER,JON BERNTHAL,8.4
10,GAME OF THRONES,SOPHIE TURNER,8.3
11,VIKINGS,KATHERYN WINNICK,8.1
12,DAREDEVIL,CHARLIE COX,7.5
13,HOW I MET YOUR MOTHER,NEIL PATRICK HARRIS,7.5
14,SUITS,PATRICK J. ADAMS,7.5
15,PEAKY BLINDERS,CILLIAN MURPHY,6.9
16,PRISON BREAK,WENTWORTH MILLER,4.2
17,THE OFFICE,RAINN WILSON,2.13333
18,FRIENDS,JENNIFER ANISTON,2.1
19,THE QUEENS GAMBIT,ANNA TAYLOR JOY,1.5
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add the series to the favorite list
```

4. Print the favorite list
5. Exit

Enter your choice: ***2***

Please enter the minimum rating: ***-1***
This is not a valid rating!

MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit

Enter your choice: ***2***

Please enter the minimum rating: ***11***
This is not a valid rating!

MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit

Enter your choice: ***2***

Please enter the minimum rating: ***5***
RANK,NAME,ACTOR,POINTS
1,DARK,LOUIS HOFMANN,9.8
2,THE WITCHER,HENRY CAVILL,9.75
3,SEINFELD,JERRY SEINFELD,9.55
4,THE WALKING DEAD,NORMAN REEDUS,9.55
5,LUCIFER,TOM ELLIS,8.9
6,BREAKING BAD,BRYAN CRANSTON,8.7
7,MODERN FAMILY,SOFIA VERGARA,8.6
8,COMMUNITY,GILLIAN JACOBS,8.5
9,THE PUNISHER,JON BERNTHAL,8.4
10,GAME OF THRONES,SOPHIE TURNER,8.3
11,VIKINGS,KATHERYN WINNICK,8.1
12,DAREDEVIL,CHARLIE COX,7.5
13,HOW I MET YOUR MOTHER,NEIL PATRICK HARRIS,7.5
14,SUITS,PATRICK J. ADAMS,7.5
15,PEAKY BLINDERS,CILLIAN MURPHY,6.9

MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit

Enter your choice: ***2***

Please enter the minimum rating: **7.5**
RANK,NAME,ACTOR,POINTS
1,DARK,LOUIS HOFMANN,9.8
2,THE WITCHER,HENRY CAVILL,9.75
3,SEINFELD,JERRY SEINFELD,9.55
4,THE WALKING DEAD,NORMAN REEDUS,9.55
5,LUCIFER,TOM ELLIS,8.9
6,BREAKING BAD,BRYAN CRANSTON,8.7
7,MODERN FAMILY,SOFIA VERGARA,8.6
8,COMMUNITY,GILLIAN JACOBS,8.5
9,THE PUNISHER,JON BERNTHAL,8.4
10,GAME OF THRONES,SOPHIE TURNER,8.3
11,VIKINGS,KATHERYN WINNICK,8.1
12,DAREDEVIL,CHARLIE COX,7.5
13,HOW I MET YOUR MOTHER,NEIL PATRICK HARRIS,7.5
14,SUITS,PATRICK J. ADAMS,7.5

MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit

Enter your choice: ***5***

Sample Run 2

```
Welcome to Netflix TV Series Ratings Calculator!
Please enter a filename for Series-Actor file: series_actors.txt
Please enter a filename for Rating file: ratings.txt
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit
---
Enter your choice: 4
---
Your favorite list is currently empty!
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit
---
Enter your choice: 3
---
Please enter the name of series: gotham
There is no such TV series!
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit
---
Enter your choice: 3
---
Please enter the name of series: the office
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
```

```
5. Exit
---
Enter your choice: 3
---
Please enter the name of series: seinfeld
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit
---
Enter your choice: 3
---
Please enter the name of series: the witcher
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit
---
Enter your choice: 4
---
RANK,NAME,ACTOR,POINTS
1,THE WITCHER,HENRY CAVILL,9.75
2,SEINFELD,JERRY SEINFELD,9.55
3,THE OFFICE,RAINN WILSON,2.13333
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit
---
Enter your choice: 5
---
```

Sample Run 3

```
Welcome to Netflix TV Series Ratings Calculator!
Please enter a filename for Series-Actor file: series_actors.txt
Please enter a filename for Rating file: ratings.txt
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit
---
Enter your choice: 2
---
Please enter the minimum rating: 8.25
RANK,NAME,ACTOR,POINTS
1,DARK,LOUIS HOFMANN,9.8
2,THE WITCHER,HENRY CAVILL,9.75
3,SEINFELD,JERRY SEINFELD,9.55
4,THE WALKING DEAD,NORMAN REEDUS,9.55
5,LUCIFER,TOM ELLIS,8.9
6,BREAKING BAD,BRYAN CRANSTON,8.7
7,MODERN FAMILY,SOFIA VERGARA,8.6
8,COMMUNITY,GILLIAN JACOBS,8.5
9,THE PUNISHER,JON BERNTHAL,8.4
10,GAME OF THRONES,SOPHIE TURNER,8.3

---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit
---
Enter your choice: 3
---
Please enter the name of series: LucIfer
---
MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit
```

Enter your choice: ***3***

Please enter the name of series: ***lucifer***
Your favorite list already have LUCIFER

MENU
1. Print the series
2. Search the series according to the rating
3. Add the series to the favorite list
4. Print the favorite list
5. Exit

Enter your choice: ***4***

RANK,NAME,ACTOR,POINTS
1,LUCIFER,TOM ELLIS,8.9

MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit

Enter your choice: ***6***

This is not a valid option!

MENU
1. Print the series
2. Search the series according to the rating
3. Add series to the favorite list
4. Print the favorite list
5. Exit

Enter your choice: ***3***

Please enter the name of series: ***californication***
There is no such TV series!

MENU
1. Print the series
2. Search the series according to the rating
3. Add the series to the favorite list
4. Print the favorite list

```
5. Exit
---
Enter your choice: 5
---
```

IMPORTANT!

If your code does not compile, then you will get **zero**. Please be careful about this and double check your code before submission.

VERY IMPORTANT!

Your programs will be compiled, executed and evaluated automatically; therefore, you should definitely follow the rules for prompts, inputs and outputs. See **Sample Runs** section for some examples.

- **Order of inputs and outputs** must be in the mentioned format.

Following these rules is crucial for grading, otherwise our software will not be able to process your outputs and you will lose some points in the best scenario.

General Rules and Guidelines about THE's

The following rules and guidelines will be applicable to all take-home exams, unless otherwise noted.

How to get help?

You can use GradeChecker (http://learnt.sabanciuniv.edu/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

You may ask questions to TAs (Teaching Assistants) or LAs (Learning Assistants) of CS201. Office hours of TAs/LAs are at the course website.

What and Where to Submit

When submitting your Take-Home Exam SUCourse, you should submit all your cpp and header files.

You should prepare (or at least test) your program using MS Visual Studio 2012 C++ (Windows users) or using Xcode (macOS users).

It'd be a good idea to write your name and last name in the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even in comment parts). If your name and last name is "Gülşen Demiröz", and if you want to write it as comment; then you must type it as follows:

// Gulsen Demiroz

Submission guidelines are below. Since the grading process will be automatic, students are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be 0.

- Name your submission file as follows:
  - Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
  - Name your cpp file that contains your program as follows: "**SUCourseUsername_THEnumber.cpp**"
  - Your SUCourse user name is actually your SUNet username, which is used for checking sabanciuniv emails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name (**use only lowercase letters**). For example, if your SUCourse username is "**altop**", then the file name should be: **altop_the4*.cpp*** (please only use lowercase letters).
  - **Please submit all .cpp and header (.h) files you used during the take-home exam (ex: strutils prompt date randgen etc.).**
  - Do not add any other character or phrase to the file name.
- Please make sure that this file is the latest version of your take-home exam program.

- Submit your work **through SUCourse only**! You can use GradeChecker only to see if your program can produce the correct outputs both in the correct order and in the correct format. It will not be considered as the official submission. You must submit your work to SUCourse. You will receive no credits if you submit by any other means (email, paper, etc.).

- If you would like to resubmit your work, you should first remove the existing file(s). This step is very important. If you do not delete the old file(s), we will receive both files and the old one may be graded.

Grading, Review and Objections

Be careful about the automatic grading: Your programs will be graded using an automated system. Therefore, you should follow the guidelines on the input and output order. Moreover, you should also use the same text as given in the "Sample Runs" section. Otherwise, the automated grading process will fail for your take-home exam, and you may get a zero, or in the best scenario, you will lose points.

Grading:

- There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse time and your computer time may have 1-2 minutes differences. You need to take this time difference into consideration.
- Successful submission is one of the requirements of the take-home exam. If, for some reason, you cannot successfully submit your take-home exam and we cannot grade it, your grade will be 0.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please submit your **own** work only. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the Syllabus or on the course website.

Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your take-home exam from the email address provided in the comment section of your announced take-home exam grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the take-home exam tab to see the problem with your take-home exam.
- Download the file you submitted to SUCourse and try to compile it.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

Good Luck!
Anıl Özdemir & Barış Altop & Gülşen Demiröz