Programming Assignment 4: IMDB

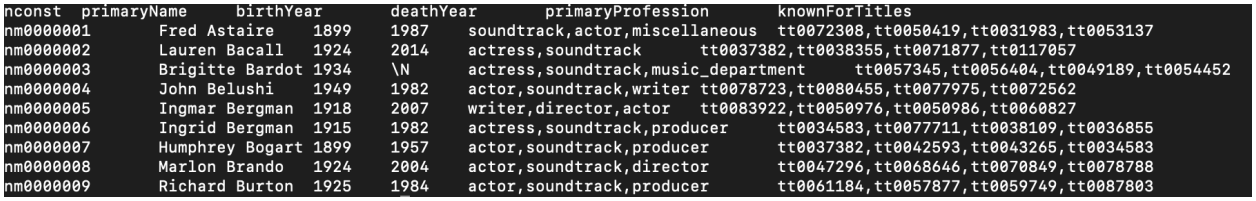
In this assignment, we want to create a simple data exploring program that can load the information from several data files and provide us with a simple interface to retrieve information from these files.
Here we have three tab separated data files that have informations about movies, actors/actresses, and cast:
1) movie.titles.tsv

tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeMinutes	genres
tt0000502	movie	Bohemios	Bohemios	0	1905	\N	100	\N
tt0000574	movie	The Story of the Kelly Gang	The Story of the Kelly Gang	0	1906	\N	70	Action,Adventure,Biography
tt0000591	movie	The Prodigal Son	L'enfant prodigue	0	1907	\N	90	Drama
tt0000615	movie	Robbery Under Arms	Robbery Under Arms	0	1907	\N	\N	Drama
tt0000630	movie	Hamlet Amleto	0	1908	\N	\N	Drama	
tt0000675	movie	Don Quijote	Don Quijote	0	1908	\N	\N	Drama
tt0000679	movie	The Fairylogue and Radio-Plays	The Fairylogue and Radio-Plays	0	1908	\N	120	Adventure,Fantasy
tt0000793	movie	Andreas Hofer	Andreas Hofer	0	1909	\N	\N	Drama
tt0000814	movie	La bocana de Mar Chica	La bocana de Mar Chica	0	1909	\N	\N	\N

This file has information about the movies. The first row is the header of this datafile that describes the information of each column. From this datafile, we care about the tconst, which is **title id**, **primaryTitle**, **startYear**, and **genres**.

You need to define an appropriate struct for **TitleRecord** and a function to read all these lines (except header) into a vector of TitleRecord: **vector<TitleRecord>**

2) movie.names.tsv



nconst	primaryName	birthYear	deathYear	primaryProfession	knownForTitles
nm0000001	Fred Astaire	1899	1987	soundtrack,actor,miscellaneous	tt0072308,tt0050419,tt0031983,tt0053137
nm0000002	Lauren Bacall	1924	2014	actress,soundtrack	tt0037382,tt0038355,tt0071877,tt0117057
nm0000003	Brigitte Bardot	1934	\N	actress,soundtrack,music_department	tt0057345,tt0056404,tt0049189,tt0054452
nm0000004	John Belushi	1949	1982	actor,soundtrack,writer	tt0078723,tt0080455,tt0077975,tt0072562
nm0000005	Ingmar Bergman	1918	2007	writer,director,actor	tt0083922,tt0050976,tt0050986,tt0060827
nm0000006	Ingrid Bergman	1915	1982	actress,soundtrack,producer	tt0034583,tt0077711,tt0038109,tt0036855
nm0000007	Humphrey Bogart	1899	1957	actor,soundtrack,producer	tt0037382,tt0042593,tt0043265,tt0034583
nm0000008	Marlon Brando	1924	2004	actor,soundtrack,director	tt0047296,tt0068646,tt0070849,tt0078788
nm0000009	Richard Burton	1925	1984	actor,soundtrack,producer	tt0061184,tt0057877,tt0059749,tt0087803

This file has information about actors/actresses. The first row is the header of this datafile that describes the information of each column. From this file we need the first five columns. The first column is our **name id**, second column is their **primaryName**, third and fourth columns are their **birthYear** and **deathYear**, respectively, and finally the fifth column is their **primaryProfession**.

You need to define an appropriate struct for **NameRecord** and a function to read all these lines (except header) into a vector of NameRecord: **vector<NameRecord>**

3) movie.principals.tsv

```
tconst  ordering        nconst  category        job     characters
tt0000502       1       nm0215752       actor   \N      \N
tt0000502       2       nm0252720       actor   \N      \N
tt0000502       3       nm0063413       director        \N      \N
tt0000502       4       nm0657268       writer  \N      \N
tt0000502       5       nm0675388       writer  \N      \N
tt0000574       10      nm0675239       cinematographer director of photography \N
tt0000574       1       nm0846887       actress \N      ["Kate Kelly"]
tt0000574       2       nm0846894       actor   \N      ["School Master"]
tt0000574       3       nm1431224       actor   \N      ["Joe Byrne"]
```

This file contains the information that shows which actor/actress plays a role in which movie (it has other information but we only care about the casting members). The first column is the **title id,** the third column is the **name id.** These two columns related the records from the first file to the records of the second file. The last column is the **character** played by the actor/actress in the corresponding movie.

You need to define an appropriate struct for **PrincipalRecord** and a function to read all these lines (except header) into a vector of PrincipalRecord: **vector<PrincipalRecord>**

The main program has two functionalities 1) searching for movies and printing the casting member associated with a selected movie 2) searching for actor/actress and printing the associated movies for them.
The searches are in lower case. The search phrases are separated with '+' and the result must contain all the indicated search phrase ('+' works like **and**)
For searching the movies the search is based on the **primaryTitle** and for actor/actress the search is based on the **primaryName.**

The first option of the menu is to search for movies. The user will be prompted to enter the search phrases (separated by '+'). The problem returns the records of all movies whose titles match all the search phrases. For each movie the program shows the index of the returned results, primaryTitle, and startYear. The search should respect the order of records in the data file, e.g., if two titles match the search phrase, the program assigns zero index to the record that appeared sooner in the data file. This is fundamental to get unique results for autograding. You can compare your output with the sample output to make sure the order is correct.
After showing the matched records, the user is prompted to select one record to see its cast members. The program shows their primaryName and their character.

```
Select a menu option:
   1. Search for movies
   2. Search for actors/actresses
   3. Exit
Your choice --> 1
Enter search phrase: god+father+II

0:
Title: The Godfather: Part II
Year: 1974
Genre: Crime Drama
---------------
1:
Title: The Godfather: Part III
Year: 1990
Genre: Crime Drama
---------------
Select a movie to see its actors/actresses (-1 to go back to the previous menu): 1
Al Pacino ["Michael Corleone"]
Diane Keaton ["Kay Adams"]
Andy Garcia ["Vincent Mancini"]
Talia Shire ["Connie Corleone Rizzi"]
```

The second menu option searches for actors/actresses. Similarly the user is prompted to enter the search phrases, and the program returns the list of cast members whose primaryName match the search phrases (note that the returned records is not inclusive to actors/actresses and show all entities from our name records).
Then the user is prompted to select a record and the program shows all movies that the selected entity has appeared on. The program shows the primaryTitle, startYear, and character.

```
Select a menu option:
   1. Search for movies
   2. Search for actors/actresses
   3. Exit
Your choice --> 2
Enter search phrase: scarlett+johan
---------------
0:
Scarlett Johansson
actress, soundtrack, producer,
(1984-\N)
---------------
Select an actor/actress to see movies (-1 to go back to the previous menu): 0
Manny & Lo 1996 ["Amanda"]
Ghost World 2001 ["Rebecca"]
My Brother the Pig 1999 ["Kathy Caldwell"]
An American Rhapsody 2001 ["Suzanne - at 15"]
Eight Legged Freaks 2002 ["Ashley Parker"]
The Perfect Score 2004 ["Francesca"]
Girl with a Pearl Earring 2003 ["Griet"]
Lost in Translation 2003 ["Charlotte"]
A Love Song for Bobby Long 2004 ["Pursy Will"]
A Good Woman 2004 ["Meg Windermere"]
In Good Company 2004 ["Alex"]
The Black Dahlia 2006 ["Kay Lake"]
The Island 2005 ["Jordan Two Delta","Sarah Jordan"]
Match Point 2005 ["Nola Rice"]
Scoop 2006 ["Sondra Pransky"]
The Other Boleyn Girl 2008 ["Mary Boleyn"]
The Prestige 2006 ["Olivia Wenscombe"]
The Nanny Diaries 2007 ["Annie Braddock"]
Vicky Cristina Barcelona 2008 ["Cristina"]
The Spirit 2008 ["Silken Floss"]
The Avengers 2012 ["Natasha Romanoff","Black Widow"]
Hitchcock 2012 ["Janet Leigh"]
The Avengers Assemble Premiere 2012 ["Self"]
Ghost in the Shell 2017 ["Major"]
Bride \N \N
We Bought a Zoo 2011 ["Kelly Foster"]
Asteroid City 2022 \N
Under the Skin 2013 ["The Female"]
Her 2013 ["Samantha"]
Captain America: The Winter Soldier 2014 ["Natasha Romanoff","Black Widow"]
Project Artemis \N \N
My Mother's Wedding \N \N
Don Jon 2013 ["Barbara"]
Little Shop of Horrors \N ["Audrey"]
Jojo Rabbit 2019 ["Rosie"]
Lucy 2014 ["Lucy"]
Chef 2014 ["Molly"]
Sing 2016 ["Ash"]
Black Widow 2021 ["Natasha Romanoff","Black Widow"]
Captain America: Civil War 2016 ["Natasha Romanoff","Black Widow"]
Broadway: The Next Generation \N ["Self"]
Rough Night 2017 ["Jess"]
Tower of Terror \N \N
American Express Unstaged: Ellie Goulding 2015 \N
Sing 2 2021 ["Ash"]
Reflective Light \N \N
Marriage Story 2019 ["Nicole Barber"]
Enter search phrase (type done to go back to the previous memu):
```

If the search phrases do not match any result the program returns “No match found!”:

```
Select a menu option:
   1. Search for movies
   2. Search for actors/actresses
   3. Exit
Your choice --> 1
Enter search phrase: god+father+IIII
No match found!

Select a menu option:
   1. Search for movies
   2. Search for actors/actresses
   3. Exit
Your choice --> 2
Enter search phrase: nicole+lowrence
No match found!
```

Notes:
Make sure that you check cin.fail() after every call to cin. In the case of cin.fail() your code should exit the loop that reads the input. This is essential for the autograder to work properly.

For example:

```
string name;
while (name != “done”) {
        cout << “enter name: ”;
        cin >> name;
        If (cin.fail() ) {
                break;
        }
        ….
}
```