Prep C
Program to access Movies database

Dr. Madhav Rao

Revision Date: July 21, 2016

Movie Database

How much digital data will the world create and store this year? Some place that number over 2 zettabyes! That's 21 zeros and is roughly the amount of storage of 57.5 billion 32GB Apple Ipads. Furthermore, the amount of data created and stored doubles every two years. How do we store and access this information? Usually with storage software known as a *database*. Databases are implemented everywhere from web servers to automobiles, video games, smartphones, etc. The goal of this project is to implement a simple C database and query system.

Tasks for your program:

- Display Movies
- Display longest movie
- Display shortest movie
- Display movies earlier than specified year
- Display movies later than specified year
- Display movies in the region of years specified
- Display movies with specified rating
- Add Movie

Program behavior

Your program should use a menu driven interface that allows the user to interact with one table in a database (see *Reading a database* for more information on a table). Once your program is launched, it should read in the file of movie information and then display the following menu:

```
Welcome to the C Movie Database!
Main menu:
    1:  display all movies
    2:  display shortest movie
    3:  display longest movie
    4:  display older movies
    5:  display newer movies
    6:  display movies in the region specified
    7:  add movie
    0:  quit the program
Enter option number:
```

The menu code (ideally a function call), should be placed in a loop. Only when the user enters a 0 should the loop be exited.

Option 1: Display all movies

Entering "1" will display the entire contents of the movie database. Based on the test *data* file, when you select the first option, your program should display something similar to:

```
The-Matrix Keanu-Reeves 90 1999
Pulp-Fiction John-Travolta 100 1994
Tombstone Kurt-Russell 100 1993
Blade-Runner Harrison-Ford 110 1982
Alien Sigourney-Weaver 95 1979
The-Shawshank-Redemption Tim-Robbins 115 1994
The-Wizard-of-Oz Judy-Garland 120 1939
A-Beautiful-Mind Russell-Crowe 180 2007
```

Option 2: Display shortest movie

This option displays the shortest movie in the database. You will need to search your list of movies to find the shortest one. Note that your program will be tested on other databases.

Option 3: Display longest movie

Like Option 2, but displays the longest movie.

Option 4: Display older movies

When the user selects this option, the program queries the user for a year:

```
Enter option number: 4
Display movies older than what year? 1940
...
```

The program then displays movies that are older than the given year.

Option 5: Display newer movies

Like Option 4:

```
Enter option number: 5
Display movies newer than what year? 1998
...
```

but movies newer than the given year are displayed.

Option 6: Display movies in the region of years specified

Like Option 5:

```
Enter option number: 6
Display movies in between what years? 1998 2011
...
```

and movies in between these years (including the years mentioned) are displayed.

Option 6: Add a movie

If you implement this option, your program should prompt the user as follows:

```
Enter option number: 6
Adding movie...
    Title: New-Movie
    Main actors: Michael-Jordon
    Year: 2012
    Length in minutes: 145
Movie added.
```

In the example, the user has entered the text `"New-Movie"`, and so on. Selecting Option 1 again should display the new movie along with the others.

Program organization

Create a *day5* directory off of your *clab* directory and move into *day5*. Your main function should be in *movies.c* and all other functions should be defined in *dbase.c*. The following sample code is given for *dbase.c*

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

int menuSelection()
    {
    int option;
    printf("..............\n");
    printf("Select an option\n");
    printf("..............\n");
    printf("1: Movie display\n");
    printf("2: Shortest movie display\n");
    printf("3: Longest movie display\n");
    ........................
    ........................
    ........................
    scanf("%d",&option);
    return option;
    }

void displayMovies(FILE *fileName)
    {
    ....
    ....
    }


void shortMovie(FILE *fileName)
    {
    ....
    ....
    }
....
....
```

```
....
....
```

The following sample code of *movies.c* is given.

```
int main(int agrc, char *argv[])
    {
    int option=-1;
    FILE *s;
    while(option)
        {
        choice = menuSelection();
        switch(choice)
            {
            ....
            ....
            ....
            ....
            }
        }
    return 1;
    }
```

Compliance Instructions

Make your *movies.dat* file in the directory *day5*. Copy-paste the movies list with other parameters mentioned above into the *movies.dat*. You should then be able to compile by running the following commands:

```
gcc movies.c dbase.c -o movies
./movies movies.dat
```

Implementation strategies

Start out by writing a program that can read and print out the database using *fscanf* and *printf* calls. Once you can successfully read and print the database, implement a menu system that handles the 0 and 1 options. With that working, implement each remaining option in turn, testing thoroughly before proceeding on to the next option.

Demonstration

Change to the *day5* directory containing your assignment. Do an *ls* command. You should see something like this:

```
dbase.c movies.c and movies.dat
```