Movie Review System Implementation

This project deals with data querying and report generation from a database that holds information about movies and viewer reviews.

- Movie data is stored in the file: moviesData.txt
- Viewer reviews are stored in the file: votingData.txt

Movie Data Structure (moviesData.txt)

Each movie is represented as follows:

Format: m_id,movie_name,Genre,Lead Studio,Year

Example content:

1,West Side Story,Drama,Disney,2021

2,Cinderella,Romance,Sony Pictures,2021

3,Last Summer,Romance,Netflix,2019

Viewer Vote Data Structure (votingData.txt)

Each vote is formatted as:

Format: m_id:vote:country:comment

Use '-' to indicate an empty comment.

Example content:

1:8:UK:Very enjoyable

2:9:USA:-

3:10:Spain:Brilliant movie

Project Structure

The main program will:

- Define the necessary data structures
- Read data from files
- Present a main menu allowing interaction with the database according to assignment requirements

Guidelines:

- Do not print inside functions unless explicitly instructed
- Each function must be documented with a comment explaining its purpose

Part A

A. Define a structure movie with the following fields:

- id: Movie identifier (positive integer)
- p2name: Pointer to the movie name (char*)
- p2genre: Pointer to the movie genre (char*)
- studio: Studio name (up to 30 characters including null terminator)
- year: Release year (positive integer)
- p2list: Pointer to a dynamic array of vote structures
- Nvotes: Number of votes for the movie

Note: id and p2name are unique identifiers.

The vote structure is defined as:

```
typedef struct {
    int value;          // The reviewer's rating
    char *p2comment;    // Comment
    char country[15];   // Country of origin
} vote;
```

B. Write a function countLines

- Takes a filename as input
- Returns the number of lines in the file (excluding the header)

C. Write a function FromFile2Movies

- Takes the movie file name, pointer to the movie array, and its size
- Reads all movie data (excluding votes) and fills the array

D. Write a function FromFile2Votes

- Takes the voting file name, movie array, and its size
- Reads votes from the file and fills the vote arrays for each movie
- Each movie will have a dynamic array of votes

E. Write a function addMovie

- Takes the movie array and its size
- Receives movie data from the user (without votes) and adds the movie to the array
- Updates the array size
- If the movie already exists, return 0; otherwise, add and return 1

F. Write a function addVote

- Takes a movie ID, movie array, and its size
- Receives vote data from the user and adds it to the corresponding movie
- If the vote already exists, return 0; otherwise, add and return 1

G. Write a function printMenu

- Displays a menu containing options from steps E-F and all functions from Part B
- Pressing 0 exits the program
- On exit, updated data must be written back to:
  - moviesData.txt for movies
  - votingData.txt for votes

Note: Ready-made data files are available. Add them to your project, and pass only the filename (not the full path) to the functions to simplify testing.

Part A - Main Function

- Call the functions from parts A-D
- Then display the menu from step G
- Remember to free all dynamically allocated memory
- Use helper functions and document each one with a purpose description

Part B - Additional Functionality

All functions below must be included in the main menu.

printVotes

- Input: Movie name, movie array, size
- Prints all comments and countries for the given movie
- Returns:
  - 0 if the vote list is empty
  - -1 if the movie doesn't exist
  - 1 otherwise

countGenre

- Input: Genre G, movie array, size
- Prints the names of all movies of genre G
- If no such movies exist, print an appropriate message

printValue

- Input: Vote value V, country C, movie array, size
- Prints names of all movies that received a vote of V from country C
- If no such votes exist, print an appropriate message

countCountry

- Input: Year Y, movie array, size
- Prints the number of distinct countries that submitted votes for any movie from year Y

maxByCountry

- Input: movie array, size
- Prints the country/countries whose users submitted the most non-empty comments
- If there's a tie, print all relevant countries

RecommendMovie

- Input: Vote threshold X, movie array, size
- Creates a file Recommendation.txt listing all movies with an average vote X
- Format: MovieName, Genre

Example:

West Side Story, Drama

deleteWorst

- Input: Genre G, movie array, size
- Deletes all votes with the lowest value for movies of genre G
- If no such genre exists in the array, print an appropriate message