ECE 270: Computer Methods in ECE

Assignment #5

Parsing

Hussein El-Souri

April 2, 2017

1) Statement of the Problem

We want to store information in a music library (in a library of CDs) in a CSV format according to the following
structure:

<Artist>,<Title>,<Year>,<Genre>,<Rating>,<Fan>

We will have to write the functions that will allow us to parse this information and
perform a query within it.

2) Description of Solution

a) First, we create a function called **readCdData** (1) that will read the data from a text file the text file is defined as read only so that the main data is never corrupted and return the number of files read as an integer.

b) Then we separate each line of information (each CD) through a strip function called **stripNewLineAtEOS** (2) that will detect a newline character '\n' and replace it with a null character (EOS character)'\n'. this is done through a for loop that runs across all data and an if condition.

c) Thirdly, we create a function called **getCommaIndex** (3) to get the index of the commas in each string. The index of the commas is stored in an array such that the first element of the array is 0 and the last is equal to string length

d) Now we create a function called **getSubString** (4) that gets a substring between two integers and here is where function (3) is useful.

e) Combining functions (3) and (4) allows us to create the function **parseCSVString** (5) that stores each parsed substring into an appropriate array (only one CD).

f) Looping function (5) along the number of lines read allows us to create **parseCSVMatrix** (6) that parses the entire data (all the CDs).

g) Now that all the information is parsed we can start printing the information using the **printCD** (7) function that prints only one CD given its line number (index).

h) Looping function (7) along all the indices (all number of lines) allows us

to print the entire data through the **printCdList** (8) function.

i) Being able to print any list of CDs we can start writing functions for the query. The functions **queryByArtist** (9)**, queryByTitle** (10), **queryByGenre** (11), **queryByFan** (12) all follow a similar logic in which we will use a string compare function that is built into C to compare a user inputted string of data to the string of data present in the parsed matrix when a match is found an integer array stores the index at where the match is found and a count is increased. That count is important because it will be entered as a parameter for function (8) along with the index array created from any of the functions (9), (10), (11), (12) according to what query is being performed. The only minor technicalities to worry about is using the gets/fgets function along to read an entire string of data along followed by a getchar function to grab the new line character.

j) The difficulty arises in functions **queryByYear** (13) and **queryByRating** (14) since these functions need to compare data stored as strings to data inputted by uses as integers. Here some helper functions **conv1**(15) and conv2(16) were created to convert data stored in year array and rating array as strings into integers. Then comparison can be done in functions (13) and (14) along the same data type.

3) Testing and Output.

The whole program was placed in a loop such that the query is performed as many times as the user desires. Attached bellow is a screenshot of the performance of each query.

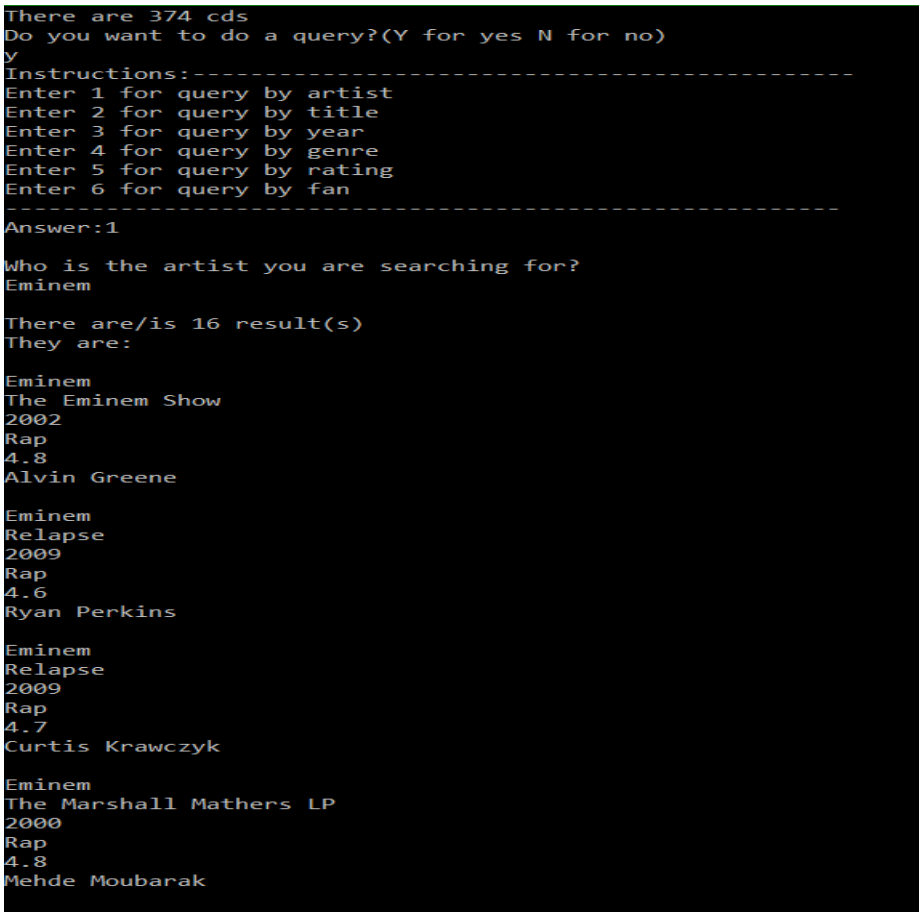

Figure 1: Query By Artist

```
Do you want to perform another query?(Y for yes N for No)
y
Instructions:---------------------------------------------
Enter 1 for query by artist
Enter 2 for query by title
Enter 3 for query by year
Enter 4 for query by genre
Enter 5 for query by rating
Enter 6 for query by fan
-------------------------------------------------------------
Answer:2

What is the title you are searching for?
Take On Me

There are/is 2 result(s)
They are:

A-ha
Take On Me
1985
Pop
4.8
Athanasios Argyris

A-ha
Take On Me
1985
Pop
4.6
Hussein El-Souri

Do you want to perform another query?(Y for yes N for No)
```

Figure 2:Query by Title

```
Enter 1 for query by artist
Enter 2 for query by title
Enter 3 for query by year
Enter 4 for query by genre
Enter 5 for query by rating
Enter 6 for query by fan
------------------------------------------------------------
Answer:3

What is the year range you are searching for?
1990 1995

There are/is 19 result(s)
They are:

Raffi
Banana Phone
1994
Vocal/Easy Listening
4.8
Bryan Brauchler

The Notorious B.I.G.
Ready To Die
1994
Hip Hop
4.6
Matthew Stachowiak

AC/DC
The Razor's Edge
1990
Hard rock
4.7
Sarah Overbeck

Fairouz
Ya Tayr
1991
muashahat
4.8
Hussein Mdaihli

Queen
Greatest Hits II
1991
Glam/Classic Rock
4.7
Collin Almario
```

Figure 3: Query by Year(Range)

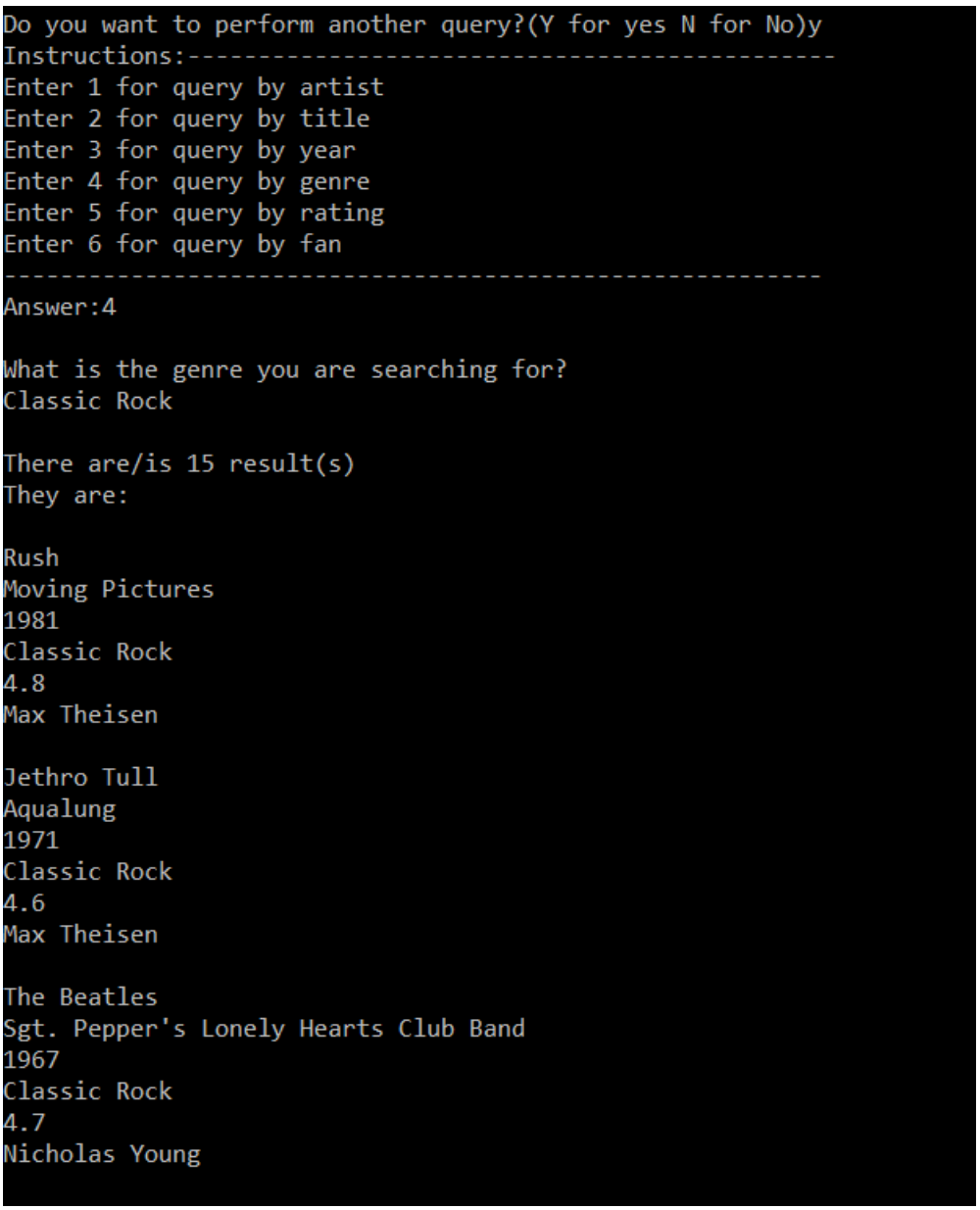

Figure 4:Query By Genre

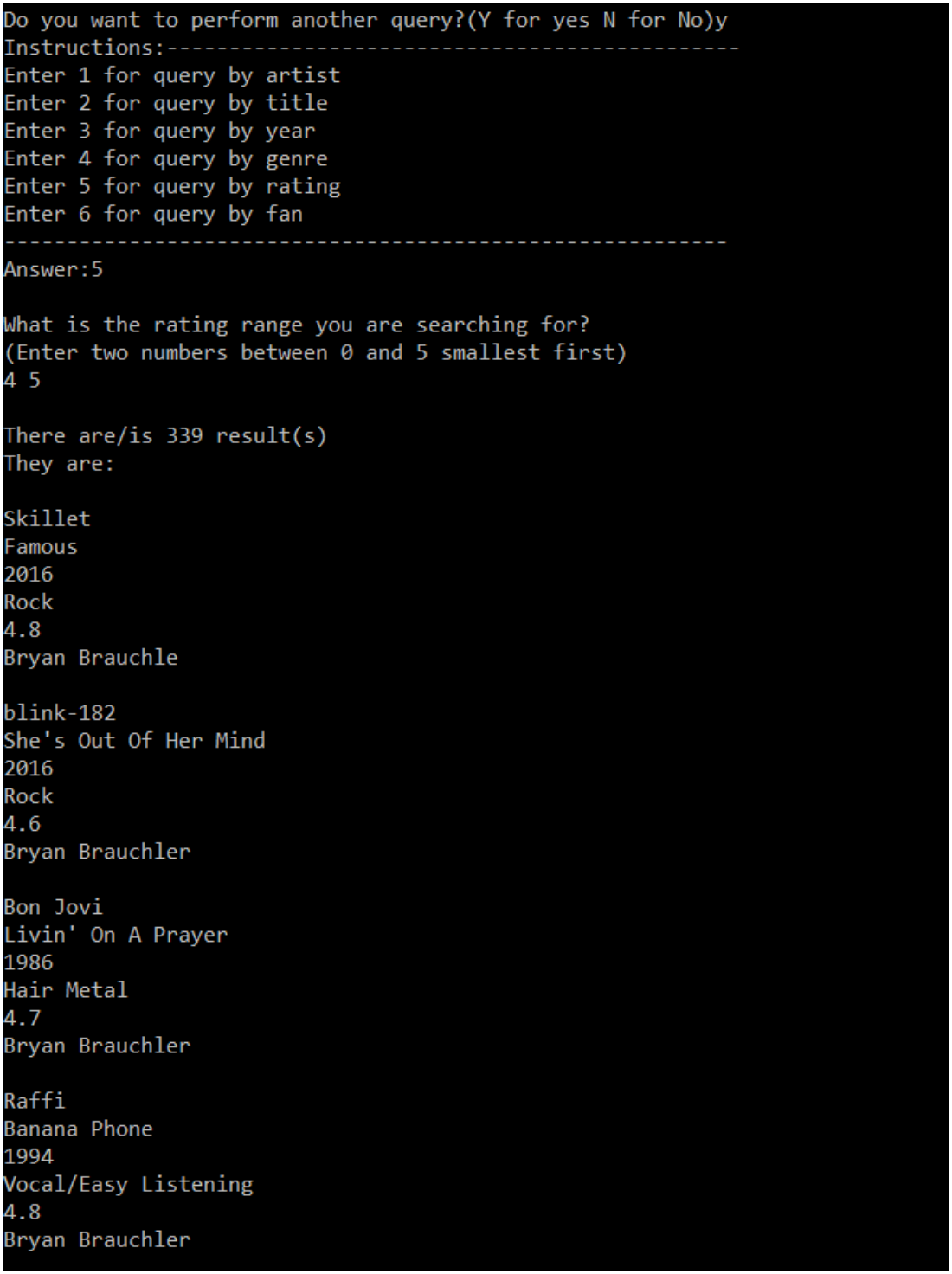

Figure 5: Query by rating
Side note result is so high because people posted music they liked and hence the ratings overall where above 4...

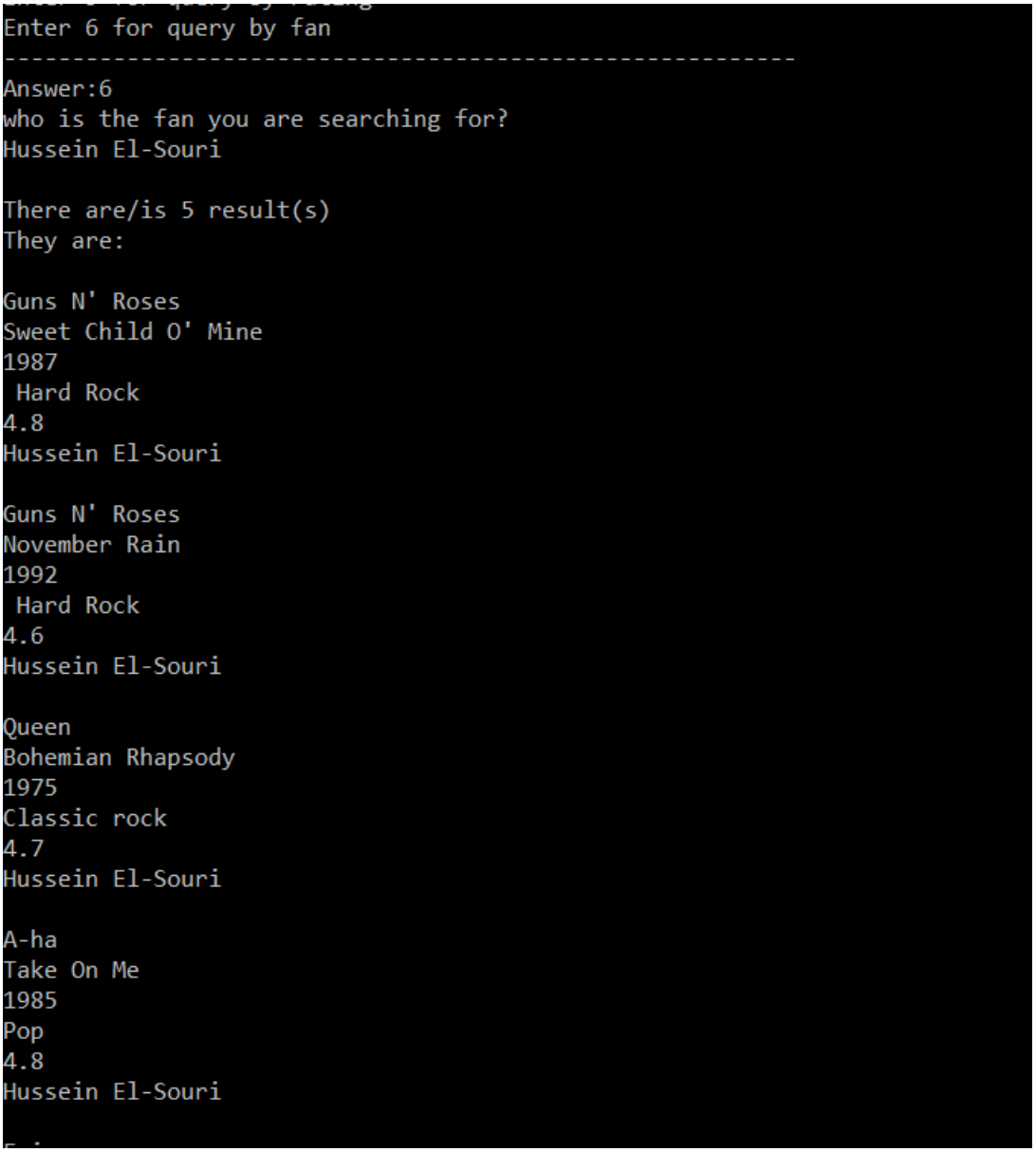

Figure 6: Query By Fan

4) Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_STR_LENGTH 400
#define MAX_NUM_CD 400

int readCdData(char filename[], char csvMatrix[][MAX_STR_LENGTH])
{
    int numLines=0;
    FILE *fp;
    fp = fopen(filename,"r");
    while( fgets(csvMatrix[numLines],MAX_STR_LENGTH,fp) != NULL)
        numLines++;
    return(numLines);
}

void stripNewLineAtEOS(char str[])
{
    int i;
    for(i=0;i<MAX_STR_LENGTH;i++)
        if (str[i]=='\n') str[i]='\0';
}

void getSubString(char str[], int i1, int i2, char substr[])
{
    int j,count=0;
    for(j=i1;j<(i2-1);j++)
    {
        substr[count]=str[j];
        count++;
    }
}

int getCommaIndex(char str[], int index[])
{
    int i,count=0,n;
    n=strlen(str);
    for (i=0; i<n; i++)
    {
        if (i==0)
        {
            index[count]=0;
            count++;
        }
        else if(str[i]==',')
        {
            index[count]=i+1;
            count++;
        }
    }
    return(count);
}

void parseCSVString(char csvString[], char artist[], char title[], char year[],
char genre[], char rating[], char fan[])
{
    int n=0,index[6],c_count=0,i;
    n=strlen(csvString);
    c_count=getCommaIndex(csvString,index);
    for(i=0;i<(c_count+1);i++)
    {
        if(i==0) getSubString(csvString,index[i],index[i+1],artist);
        else if(i==1) getSubString(csvString,index[i],index[i+1],title);
        else if (i==2) getSubString(csvString,index[i],index[i+1],year);
```

```
        else if (i==3) getSubString(csvString,index[i],index[i+1],genre);
        else if (i==4) getSubString(csvString,index[i],index[i+1],rating);
        else if (i==5) getSubString(csvString,index[i],n,fan);
    }
}

void parseCSVMatrix(int n, char csvMatrix[][MAX_STR_LENGTH], char
cdDB_artist[][MAX_STR_LENGTH],char cdDB_title[][MAX_STR_LENGTH], char
cdDB_year[][MAX_STR_LENGTH], char cdDB_genre[][MAX_STR_LENGTH], char
cdDB_rating[][MAX_STR_LENGTH], char cdDB_fan[][MAX_STR_LENGTH])
{
    int i;
    for(i=0;i<n;i++)
    {
        parseCSVString(csvMatrix[i], cdDB_artist[i], cdDB_title[i], cdDB_year[i],
cdDB_genre[i], cdDB_rating[i], cdDB_fan[i]);
    }
}

void printCD(int index, char cdDB_artist[][MAX_STR_LENGTH],char
cdDB_title[][MAX_STR_LENGTH], char cdDB_year[][MAX_STR_LENGTH], char
cdDB_genre[][MAX_STR_LENGTH], char cdDB_rating[][MAX_STR_LENGTH], char
cdDB_fan[][MAX_STR_LENGTH])
{
    printf("%s\n%-s\n%-s\n%-s\n%-s\n%-
s\n",cdDB_artist[index],cdDB_title[index],cdDB_year[index],cdDB_genre[index],cdDB_r
ating[index],cdDB_fan[index]);
}

void printCdList(int n, int index[], char cdDB_artist[][MAX_STR_LENGTH],char
cdDB_title[][MAX_STR_LENGTH], char cdDB_year[][MAX_STR_LENGTH], char
cdDB_genre[][MAX_STR_LENGTH], char cdDB_rating[][MAX_STR_LENGTH], char
cdDB_fan[][MAX_STR_LENGTH])
{
    int i;
    for (i=0;i<n;i++)
    {
        prindCd(i,cdDB_artist,cdDB_title, char cdDB_year, char cdDB_genre, char
cdDB_rating, char cdDB_fan);
    }
}

int queryByArtist(int n, char artistOfInterest[],int index[], char
cdDB_artist[][MAX_STR_LENGTH])
{
    int i,count=0,result;
    for(i=0;i<n;i++)
    {
        result=strcmp(artistOfInterest,cdDB_artist[i]);
        if(result==0)
            {
                index[count]=i;
                count++;
            }
    }
    return(count);
}

int queryByTitle(int n, char titleOfInterest[],int index[], char
cdDB_title[][MAX_STR_LENGTH])
{
    int i,count=0,result;
    for(i=0;i<n;i++)
    {
        result=strcmp(titleOfInterest,cdDB_title[i]);
        if(result==0)
        {
            index[count]=i;
```

```
            count++;
        }
    }
    return(count);
}

void conv1 (int value[],char cdDB_year[][MAX_STR_LENGTH])
{
    int i,temp=0;
    for(i=0;i<MAX_NUM_CD;i++)
    {
        temp=atoi(cdDB_year[i]);
        value[i]= temp ;
    }
}
int queryByYear(int n, int minYear, int maxYear, int index[],int cdDB_year[])
{
    int i,count=0,temp=0;
    for(i=0;i<n;i++)
    {
        temp=cdDB_year[i];
        if(temp>=minYear&&temp<=maxYear)
        {
            index[count]=i;
            count++;
        }
    }
    return(count);
}

int queryByGenre(int n, char genreOfInterest[],int index[], char
cdDB_genre[][MAX_STR_LENGTH])
{
    int i,count=0,result;
    for(i=0;i<n;i++)
    {
        result=strcmp(genreOfInterest,cdDB_genre[i]);
        if(result==0)
        {
            index[count]=i;
            count++;
        }
    }
    return(count);
}

void conv2(float value[],char cdDB_rating[][MAX_STR_LENGTH])
{
    int k;
    float temp=0;
    for(k=0;k<MAX_NUM_CD;k++)
    {
        temp=atof(cdDB_rating[k]);
        value[k]=temp;
    }
}
int queryByRating(int n, float minRating, float maxRating, int index[],float
cdDB_rating[])
{
    int i,count=0,temp=0;
    for(i=0;i<n;i++)
    {
        temp=cdDB_rating[i];
        if(temp>=minRating&&temp<=maxRating)
        {
            index[count]=i;
            count++;
        }
```

```
    }
    return(count);
}

int queryByFan(int n, char fanOfInterest[],int index[], char
cdDB_fan[][MAX_STR_LENGTH])
{
    int i,count=0,result;
    for(i=0;i<n;i++)
    {
        result=strcmp(fanOfInterest,cdDB_fan[i]);
        if(result==0)
            {
                index[count]=i;
                count++;
            }
    }
    return(count);
}


int main(void)
{
    int numLines=0, index[MAX_STR_LENGTH],answer=0,count=0,test,year1=0,year2=0;
    int year_int[MAX_NUM_CD];
    float rating1=0.0,rating2=0.0,rating_fl[MAX_NUM_CD];
    char csvMatrix[MAX_NUM_CD][MAX_STR_LENGTH],isPlaying='c';
    char artist[MAX_NUM_CD][MAX_STR_LENGTH], title[MAX_NUM_CD][MAX_STR_LENGTH],
year[MAX_NUM_CD][MAX_STR_LENGTH], genre[MAX_NUM_CD][MAX_STR_LENGTH],
rating[MAX_NUM_CD][MAX_STR_LENGTH], fan[MAX_NUM_CD][MAX_STR_LENGTH];
    char
artistOfInterest[MAX_STR_LENGTH],titleOfInterest[MAX_STR_LENGTH],genreOfInterest[MA
X_STR_LENGTH], fanOfInterest[MAX_STR_LENGTH];
    numLines=readCdData("data.txt",csvMatrix);
    printf("There are %d cds\n",numLines);
    stripNewLineAtEOS(csvMatrix);
    parseCSVMatrix(numLines,csvMatrix,artist,title,year,genre,rating,fan);
    conv1(year_int,year);
    conv2(rating_fl,rating);
    printf("Do you want to do a query?(Y for yes N for no)\n");
    scanf("%c",&isPlaying);getchar();
    while(isPlaying=='Y'||isPlaying=='y')
    {
        printf("Instructions:---------------------------------------------\n");
        printf("Enter 1 for query by artist\n");
        printf("Enter 2 for query by title\n");
        printf("Enter 3 for query by year\n");
        printf("Enter 4 for query by genre\n");
        printf("Enter 5 for query by rating\n");
        printf("Enter 6 for query by fan\n");
        printf("-------------------------------------------------------------\n");
        printf("Answer:");
        scanf("%d",&answer);getchar();
        if(answer==1)
        {
            printf("\nWho is the artist you are searching for?\n");
            gets(artistOfInterest);
            count=queryByArtist(numLines,artistOfInterest,index,artist);
            if(count!=0)
            {
            printf("\nThere are/is %d result(s)\nThey are:\n",count);
            printCdList(count,index,artist,title,year,genre,rating,fan);
            }
            else
            {
                printf("\nThere are no results matching this artist\n");
            }
            printf("\nDo you want to perform another query?(Y for yes N for
```

```
No)\n");
            scanf("%c",&isPlaying);getchar();

        }
        else if(answer==2)
        {
            printf("\nWhat is the title you are searching for?\n");
            gets(titleOfInterest);
            count=queryByTitle(numLines,titleOfInterest,index,title);
            if(count!=0)
            {
                printf("\nThere are/is %d result(s)\nThey are:\n",count);
                printCdList(count,index,artist,title,year,genre,rating,fan);
            }
            else
                printf("There are no results matching your query. :(");
            printf("\nDo you want to perform another query?(Y for yes N for No)");
            scanf("%c",&isPlaying);getchar();
        }

        else if(answer==3)
        {
            printf("\nWhat is the year range you are searching for?\n");
            scanf("%d%d",&year1,&year2);getchar();
            count=queryByYear(numLines,year1,year2,index,year_int);
            if(count!=0)
            {
                printf("\nThere are/is %d result(s)\nThey are:\n",count);
                printCdList(count,index,artist,title,year,genre,rating,fan);
            }
            else
                printf("There are no results matching your query. :(");
            printf("\nDo you want to perform another query?(Y for yes N for No)");
            scanf("%c",&isPlaying);getchar();
        }

        else if(answer==4)
        {
            printf("\nWhat is the genre you are searching for?\n");
            gets(genreOfInterest);
            count=queryByGenre(numLines,genreOfInterest,index,genre);
            if(count!=0)
            {
                printf("\nThere are/is %d result(s)\nThey are:\n",count);
                printCdList(count,index,artist,title,year,genre,rating,fan);
            }
            else
                printf("There are no results matching your query. :(\n");
            printf("\nDo you want to perform another query?(Y for yes N for No)");
            scanf("%c",&isPlaying);getchar();
        }

        else if(answer==5)
        {
            printf("\nWhat is the rating range you are searching for?\n(Enter two
numbers between 0 and 5 smallest first)\n");
            scanf("%f%f",&rating1,&rating2);
            count=queryByRating(numLines,rating1,rating2,index,rating_fl);
            if(count!=0)
            {
                printf("\nThere are/is %d result(s)\nThey are:\n",count);
                printCdList(count,index,artist,title,year,genre,rating,fan);
            }
            else
                printf("There are no results matching your query. :(");
            printf("\nDo you want to perform another query?(Y for yes N for No)");
            scanf("%c",&answer);getchar();
        }
```

```
        else if(answer==6)
        {
            printf("who is the fan you are searching for?\n");
            gets(fanOfInterest);
            count=queryByFan(numLines,fanOfInterest,index,fan);
            if(count!=0)
            {
                printf("\nThere are/is %d result(s)\nThey are:\n",count);
                printCdList(count,index,artist,title,year,genre,rating,fan);
            }
            else
                printf("There are no results matching your query. :(\n");
            printf("\nDo you want to perform another query?(Y for yes N for No)");
            scanf("%c",&isPlaying);getchar();
        }

    }
    printf("\nHave a good day. :)\n");
    return 0;
}
```