```
/*******************************************************************************
 * PROGRAMMER : Ali Eshghi
 * STUDENT ID : 1112261
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * Assign #3  : Searching linked list
 * DUE DATE   : 6 November 2019
 ******************************************************************************/

#include "Header.h"

/*******************************************************************************
 * Function - SearchRate
 * ---------------------------------------------------------------------------
 * This function will get a rating from the user and search the genre content of
 * each node to see if there is a matching rating in the nodes with the rating
 * searched by the user. if found, the function prints out every content of
 * every node for the user, if not found, the function outputs that the rating
 * searched by the user was not found.
 *
 * return type - nothing
 *              the function is void type
 ******************************************************************************/

void SearchRate(MovieNode *head, ofstream &outFile)
{
    /*************
     * VARIABLES *
     ************/

    MovieNode *movPtr;  //Proc - stores the target name
    MovieNode node;         //Proc & In - passes the info into persPtr


    int i;                  //In - counts the founded data

    int checkInt;           //In - stores input value

    bool checkInp;          //Proc - LCV value

    /****************
     * INITIALIZING *
     ****************/
    checkInp = false;

    i = 1;

    /**************************************************************************
     * PROCESSING - searching for the rating in the list one by one and output
     *              each movie with the matching rating to the output file
     **************************************************************************/


    if (head == NULL)
    {
```

```cpp
        cout << endl;

        cout << "Can not search for a rating in an empty list.";

        cout << endl;
    }

    else
    {
        //creating new node

        movPtr = new MovieNode;

        //INPUT

        cout << endl;

        do
        {
            //INPUT

            cout << "\nWhich rating are you looking for? ";

            //CHECKS FOR THE CHAR INPUT

            if (!(cin >> checkInt))
            {

                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');

                cout << endl;
                cout << "**** Please input a NUMBER between 0 and 9 ****";
                cout << endl;

                checkInp = false;


            }

            //CHECKS FOR THE RANGE ERROR

            else if (checkInt >= 10 || checkInt <= -1 )
            {

                cout << endl;
                cout << "**** The number "              << node.rate
                     << " is an invalid entry     ****" << endl;
                cout << "**** Please input a number between 0 and 9 ****";
                cout << endl;

                checkInp = false;

            }

            //PASS

            else
            {

                cin.ignore(numeric_limits<streamsize>::max(), '\n');
                checkInp = true;

            }

        }while(!checkInp);

        node.rate = checkInt;

        *movPtr = node;

        cout << endl;

        cout << "Searching for the rating";

        cout << endl;

        outFile << "\nSearch by rating for " << movPtr -> rate << " found:";

        outFile << endl << endl;

        outFile << left;

        outFile << setw(9)  << "MOVIE #"           ;
        outFile << setw(48) << "TITLE"             ;
        outFile << setw(5)  << "YEAR"              ;
        outFile << setw(8)  << "RATING"            ;
        outFile << setw(18) << "GENRE"             ;
        outFile << setw(18) << "ALT GENRE"         ;
        outFile << setw(20) << "LEAD ACTOR"        ;
        outFile << setw(20) << "SUPPORTING ACTOR"  ;

        outFile << left;

        outFile << endl;

        outFile << "------- ";
        outFile << "------------------------------------------------ ";
        outFile << "---- ";
        outFile << "------ ";
        outFile << "----------------- ";
        outFile << "----------------- ";
        outFile << "------------------- ";
        outFile << "------------------- ";
        outFile << endl;

        //GOES THROUGH THE LIST

        while(head -> next != NULL)
        {
            if (head -> rate == movPtr -> rate)
            {

            outFile << right;
            outFile << setw(4)  << i << "      "       ;
            outFile << left;
            outFile << setw(47) << head -> title       ;
            outFile << setw(8)  << head -> year        ;
            outFile << setw(5)  << head -> rate        ;
            outFile << setw(18) << head -> genre       ;
            outFile << setw(18) << head -> altGenre    ;
            outFile << setw(20) << head -> leadActor   ;
            outFile << setw(20) << head -> supportActor;
            outFile << endl;
            outFile << left;

            i++;
            head = head -> next;

            }

            else
            {
            head  = head -> next;
            }

        } // END - WHILE


        if (head -> next == NULL && (head -> rate == movPtr -> rate))
        {

            outFile << right;
            outFile << setw(4)  << i << "      "       ;
            outFile << left;
            outFile << setw(47) << head -> title       ;
            outFile << setw(8)  << head -> year        ;
            outFile << setw(5)  << head -> rate        ;
            outFile << setw(18) << head -> genre       ;
            outFile << setw(18) << head -> altGenre    ;
            outFile << setw(20) << head -> leadActor   ;
            outFile << setw(20) << head -> supportActor;
            outFile << endl;
            outFile << left;
            i++;
            head = head -> next;

        }

        //IF NOT FOUND IN THE LIST

        if (i == 1)
        {
            cout << "Sorry, no movies for the rating, " << movPtr -> rate
                 << " were found.";

            cout << endl;
        }

        //OUTPUT RESULT INTO THE CONSOLE

        if(i > 1)
        {
            cout << "Found " << i-1 << " movies for the rating "
                 << movPtr -> rate << '!' << endl;
        }

        movPtr = NULL;
        delete movPtr;

     }

}

```