

Colegiul Național „Gheorghe Șincai” Baia Mare

Lucrare în vederea obținerii atestatului profesional la informatică

Sistem de logare pentru o bibliotecă

Îndrumători,
prof. Belbe Maria
prof. Coroiu Mircea

elev Mariș Radu-Ioan
clasa a XII-a B

2021

Cuprins

Introducere.....	2
Descrierea programului	3
Ghid de utilizare	4
Meniul Client.....	8
Meniul Administrator	12
Cod Sursă.....	15
Bibliografie. Webografie.....	29

Introducere

Aplicația „Sistem de logare pentru o bibliotecă” este realizată în limbajul de programare C++. Acest limbaj permite crearea de programe în consolă.

Code::Blocks este o platformă open-source care permite dezvoltarea aplicațiilor C++ și poate fi folosită pe Windows, Linux sau macOS și care suportă o varietate de compilatoare.

Aplicația dezvoltată are un dublu rol: să ajute atât vizitatorii bibliotecii cât și persoanelor acesteia.

Ideea a pornit de la împrumutul unei cărți de la bibliotecă în timpul pandemiei. Deplasarea până la instituție era necesară, doar pentru a se putea constata faptul că anumite volume nu existau în inventarul bibliotecii. De aceea, acest program este menit să informeze oamenii în legătură cu cărțile existente în bibliotecă, fără a fi nevoie de interacțiune directă cu personalul bibliotecii.

Programul este menit să ajute la fel de mult și în cadrul instituției, fiind o modalitate de edita stocul. Astfel, o persoană autorizată poate folosi contul de administrator pentru a înregistra o carte în baza de date în cazul donării de cărți bibliotecii, dar poate să și șteargă o carte în eventualitatea în care o carte a fost distrusă.

Descrierea programului

„Sistem de logare pentru o bibliotecă” a fost realizat folosind structuri de date si biblioteci din cadrul „Code::Blocks”. Aplicația este structurată sub forma unui sistem de logare personalizat pentru fiecare utilizator. Acesta va putea interacționa cu aplicația prin intermediul consolei.

Am folosit biblioteci din cadrul „Code::Blockes” precum:

- `iostream` – include elemente din biblioteca standard C++ de intrare si ieșire
- `fstream` – permite lucrul cu fisiere, mai exact scrierea si citirea din acestea
- `conio.h` – conține funcția `getch`, care permite citirea oricărei taste, nefiind nevoie de apăsarea tastei `enter`
- `windows.h` – folosită pentru atribuirea culorilor în consolă
- `io manip` – permite folosirea de manipulări parametrice
- `cstring` – permite manipularea caracterelor și a șirurilor de caractere

Aplicația este împărțită în 2 tipuri de utilizatori:

- Client - cu acces la vizualizarea cărților din bibliotecă, afișarea lor în ordine cronologică si căutarea cărților în funcție de titlu, autor sau editură;
- Admin - are acces la toate facilitățile utilizatorului de tip Client, dar poate adăuga și șterge cărți din bibliotecă;

Aplicația permite oricărei persoane să creeze un cont de tip Client pentru a putea avea acces la permisiunile pe care le oferă, însă există un singur cont de tip Admin. Amandouă tipurile de utilizatori au acces la posibilitatea schimbării parolei.

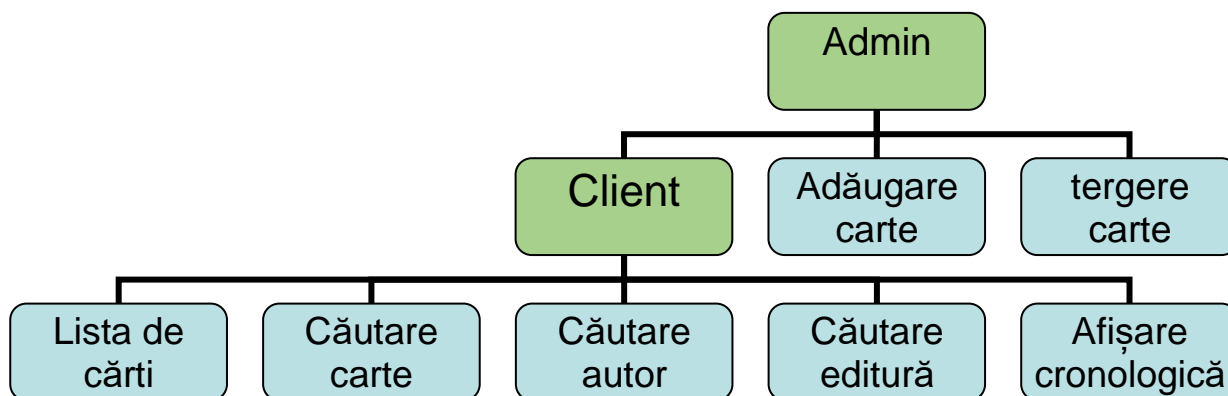


Fig.1

Ghid de utilizare

Meniul principal este prezentat la deschiderea aplicației, constând în enunțarea scopului programului prin titlu: „Biblioteca”, menit să informeze printr-un design atractiv. De asemenea sunt prezente opțiunile de logare.

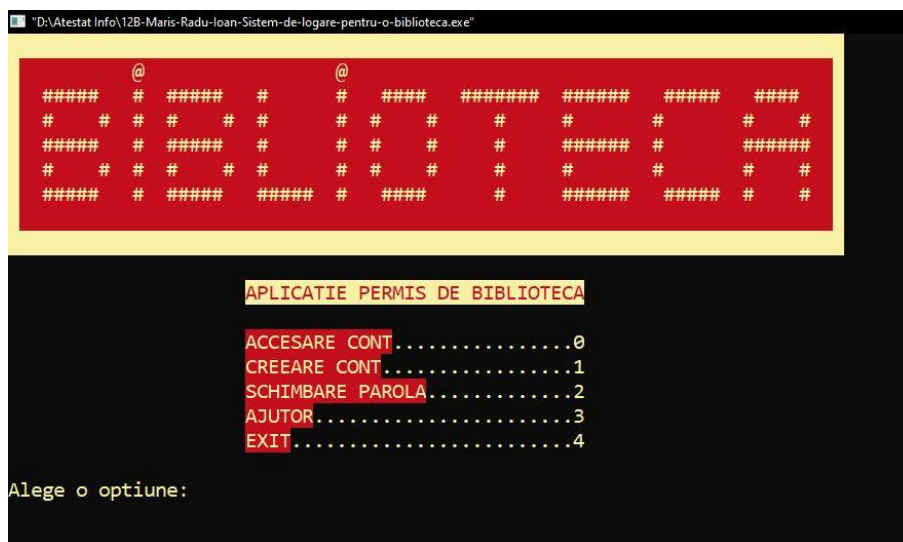


Fig.2

La apasarea tastei 0, persoana care folosește aplicația poate să se conecteze în contul lui prin introducerea numelui său de utilizator si a parolei.

În cazul in care numele de utilizator nu există sau parola este incorectă, mesajul „Numele contului sau parola sunt gresite!” este afișat pe ecran.

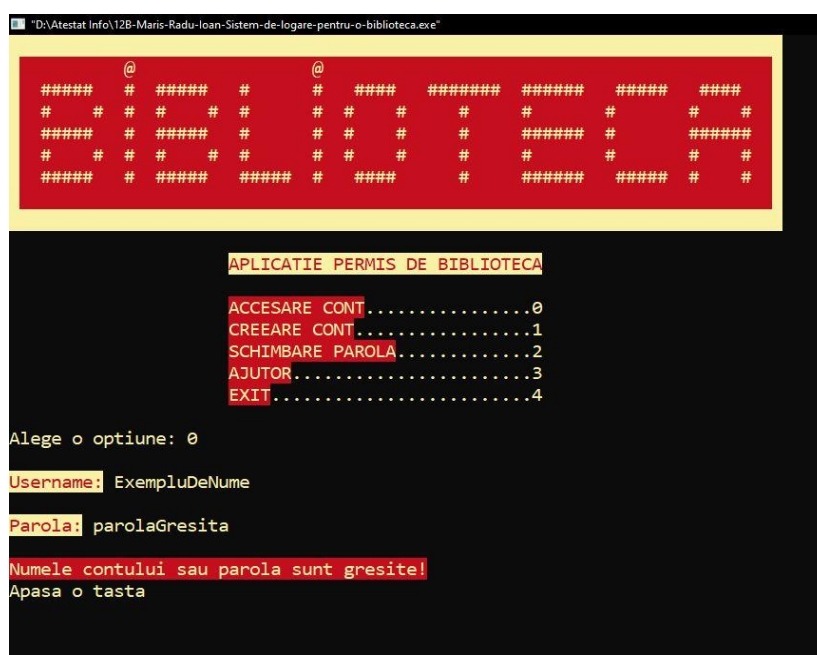
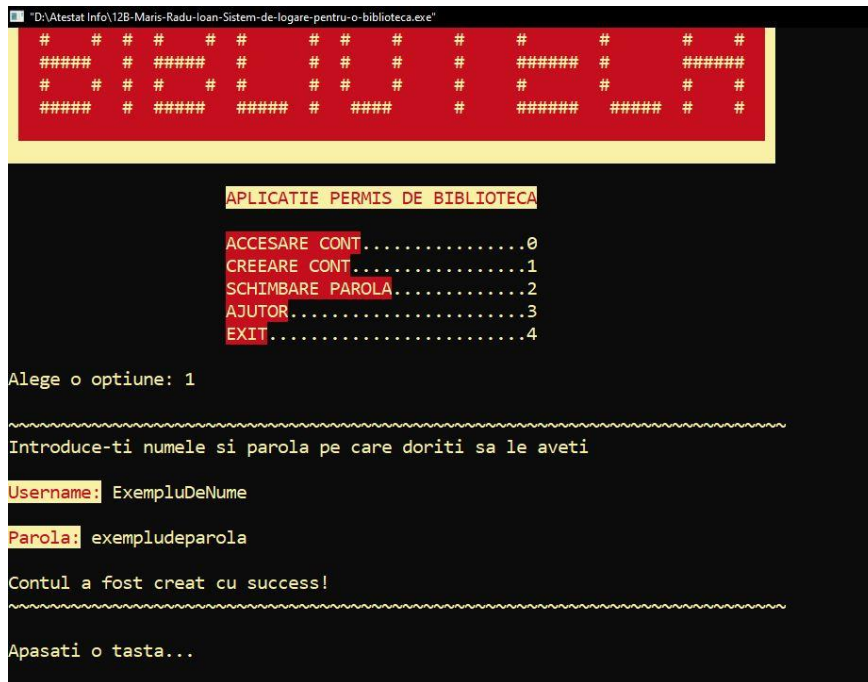


Fig.3

La apăsarea tastei 1, persoana poate să creeze un nou cont de tip client, compus din username (nume de utilizator) si parolă.



```
"D:\Atestat Info\12B-Maris-Radu-Ioan-Sistem-de-logare-pentru-o-biblioteca.exe"
#####
# # # # # # # # # # # # # # # #
##### # #####
# # # # # # # # # # # # # # # #
##### # ##### # ##### # #

APLICATIE PERMIS DE BIBLIOTECA

ACCESARE CONT.....0
CREEARE CONT.....1
SCHIMBARE PAROLA.....2
AJUTOR.....3
EXIT.....4

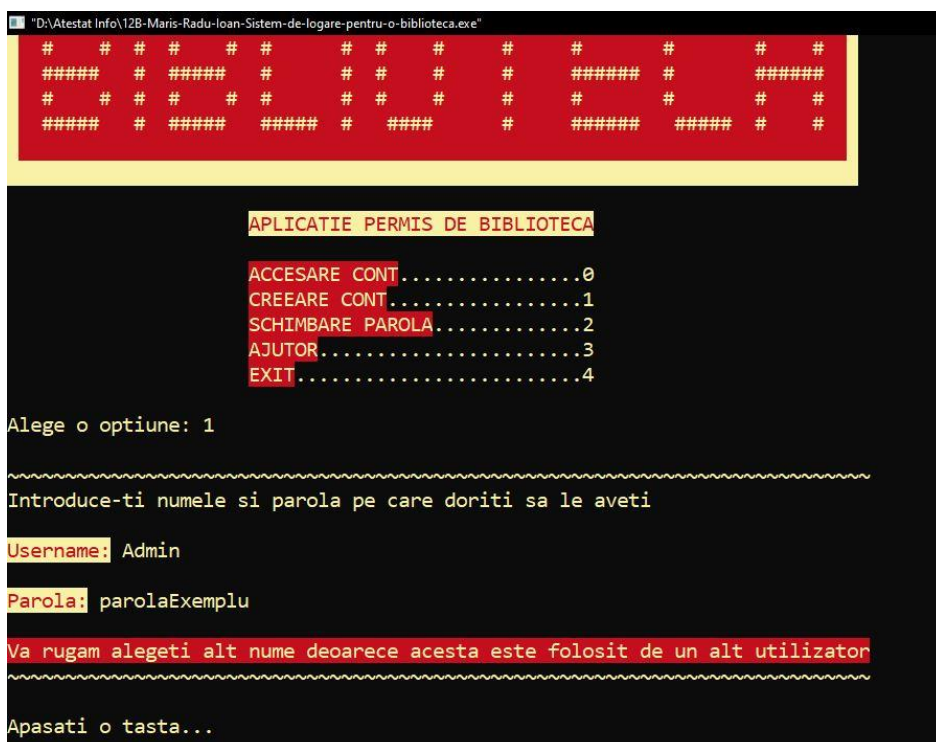
Alege o optiune: 1

~~~~~
Introduce-ti numele si parola pe care doriti sa le aveti
Username: ExempluDeNume
Parola: exempludeparola

Contul a fost creat cu success!
~~~~~
Apasati o tasta...
```

Fig.4

Cu toate acestea, nu pot exista 2 utilizatori cu același nume. În cazul în care numele ales este folosit de o altă persoană, contul nu va fi creat și programul se va întoarce la meniul principal.



```
"D:\Atestat Info\12B-Maris-Radu-Ioan-Sistem-de-logare-pentru-o-biblioteca.exe"
#####
# # # # # # # # # # # # # # # #
##### # #####
# # # # # # # # # # # # # # # #
##### # ##### # ##### # #

APLICATIE PERMIS DE BIBLIOTECA

ACCESARE CONT.....0
CREEARE CONT.....1
SCHIMBARE PAROLA.....2
AJUTOR.....3
EXIT.....4

Alege o optiune: 1

~~~~~
Introduce-ti numele si parola pe care doriti sa le aveti
Username: Admin
Parola: parolaExemplu

Va rugam alegeti alt nume deoarece acesta este folosit de un alt utilizator
~~~~~
Apasati o tasta...
```

Fig.5

La apăsarea tastei 2, persoana poate să își schimbe parola. Pentru ca acțiunea să fie cu succes, utilizatorul trebuie să își introducă numele și parola pe care dorește să o schimbe, iar mai apoi noua parolă.

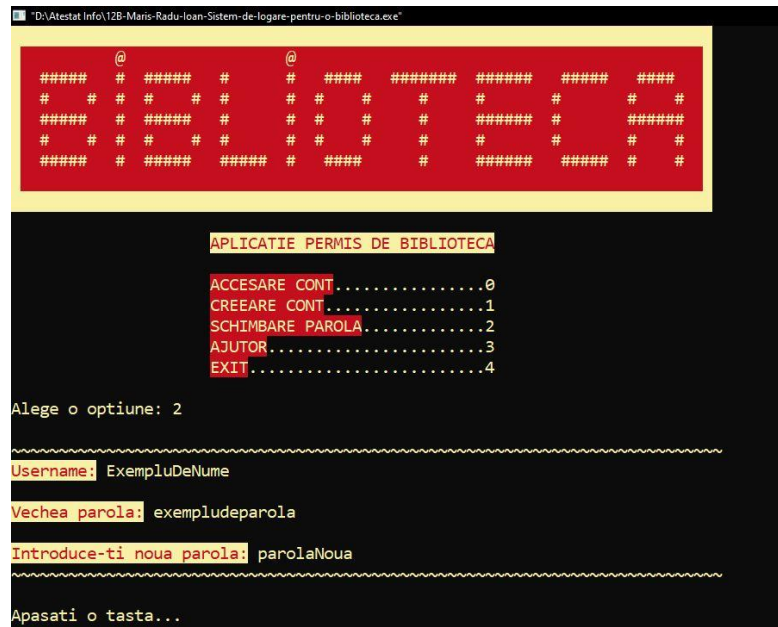


Fig.6

Există posibilitatea în care se dorește schimbarea parolei, însă utilizatorul nu există sau vechea parolă nu este corectă. În acest caz, pe ecran este afișat mesajul „Numele contului sau parola sunt gresite!”. Acesta, fiind un mesaj de eroare, este conturat cu roșu pentru a ieși în evidență.

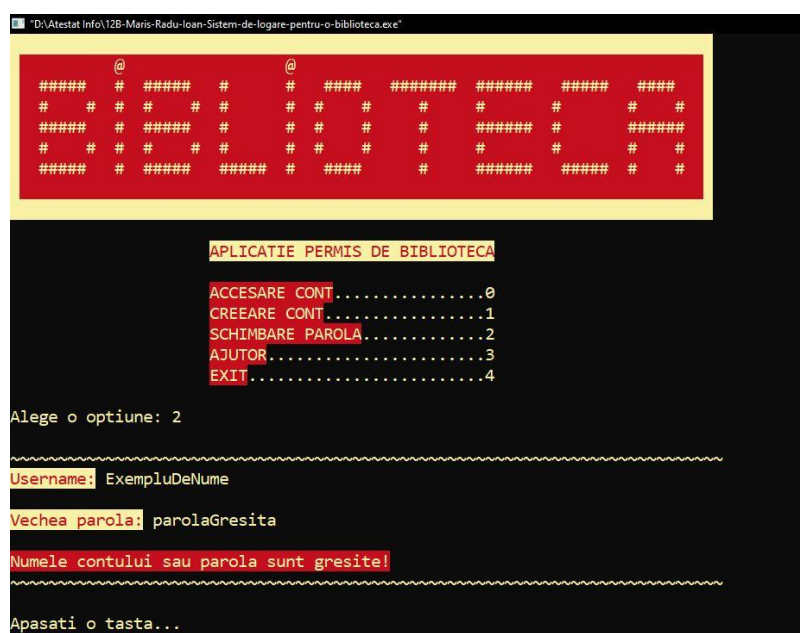
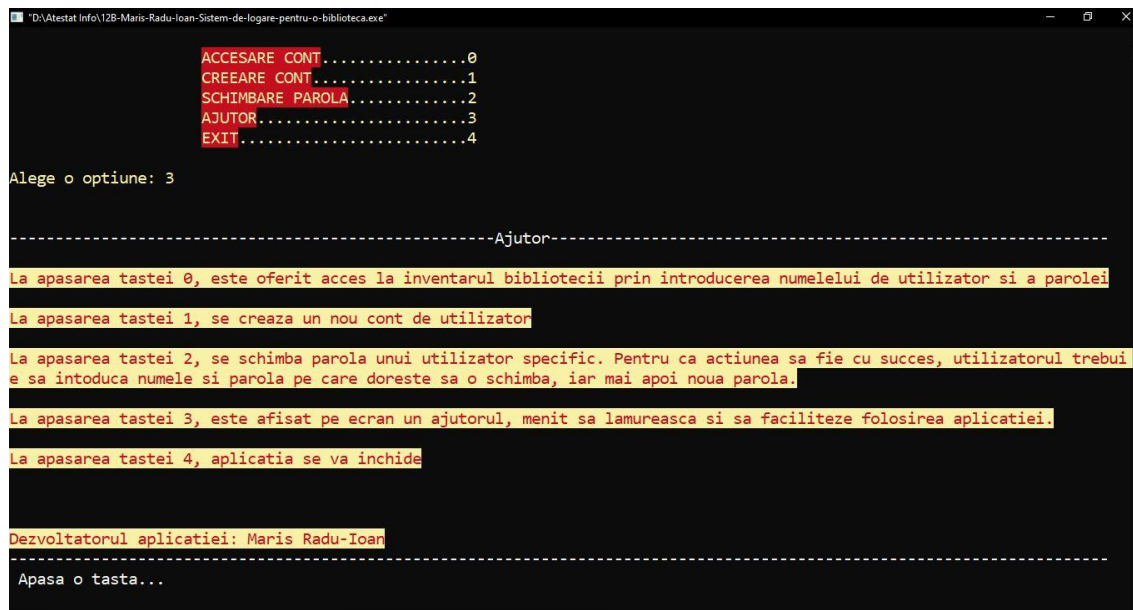


Fig.7

La apăsarea tastei 3, este afișat pe ecran un ajutor, menit să lămurească și să faciliteze folosirea aplicației.



```
"D:\Atestat Info\128-Maris-Radu-Ioan-Sistem-de-logare-pentru-o-biblioteca.exe"
ACESARE CONT.....0
CREARE CONT.....1
SCHIMBARE PAROLA.....2
AJUTOR.....3
EXIT.....4

Alege o optiune: 3

-----Ajutor-----

La apasarea tastei 0, este oferit acces la inventarul bibliotecii prin introducerea numelui de utilizator si a parolei
La apasarea tastei 1, se creaza un nou cont de utilizator
La apasarea tastei 2, se schimba parola unui utilizator specific. Pentru ca actiunea sa fie cu succes, utilizatorul trebuie sa introduca numele si parola pe care doreste sa o schimba, iar mai apoi noua parola.
La apasarea tastei 3, este afisat pe ecran un ajutorul, menit sa lamureasca si sa faciliteze folosirea aplicatiei.
La apasarea tastei 4, aplicatia se va inchide

Dezvoltatorul aplicatiei: Maris Radu-Ioan
-----
Apasa o tasta...
```

Fig.8

Tasta 4 reprezintă ieșirea din program, iar apăsarea acesteia va cauza închiderea aplicației.

Meniul Client

În cazul accesării contului, pentru un utilizator de tip Client, se va afișa pe ecran un nou meniu constând dintr-un mesaj de întâmpinare și a opțiunii disponibile:

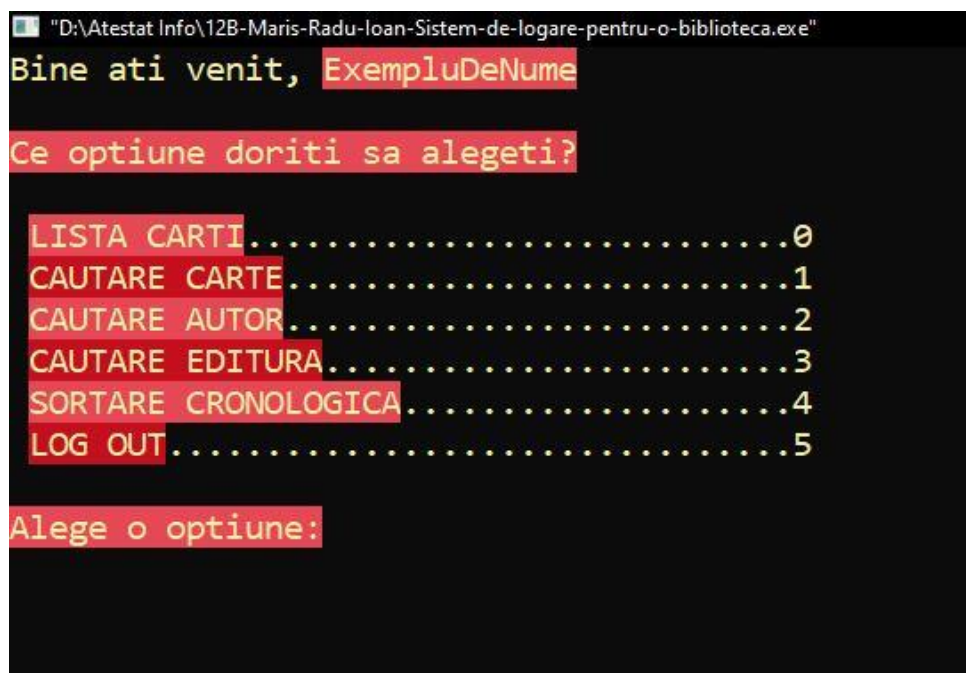


Fig.9

LISTA CARTI - La apăsarea tastei 0 afișează toate informațiile legate de cărțile care aparțin bibliotecii și toate informațiile legate de acestea pe coloane diferite.

The screenshot shows the 'LISTA CARTI' interface, which displays a table of books. The table has five columns: Titlu, Autor, Editura, An publicare, and Cod ISBN. The data is as follows:

Titlu	Autor	Editura	An publicare	Cod ISBN
Alexandru Lapusneanu	Costache Negruzzi	Unirea	1840	9574861235486
Amintiri din Copilarie	Ion Creanga	editura 2	2000	9789739987452
Cuvinte potrivite	Tudor Arghezi	Albatros	1927	9578452136542
Enigma Otiliei	George Calinescu	Recif	1938	4587695845784
Harap-Alb	Ion Creanga	Accent	1877	2154215876589
Ion	Liviu Rebreanu	editura 1	2000	9789732365478
Moara cu Noroc	Ioan Slavici	Minerva	1881	1000002154785
O scrisoare pierduta	Ion Luca Caragiale	Dacia	1884	2548754236548
Poemele luminii	Lucian Blaga	Unirea	1919	4586952365891
Volum de poezii	Mihai Eminescu	editura 1	1980	9781234567890

Apasa Enter...

Fig.10

CAUTARE CARTE - La apăsarea tastei 1 se permite introducerea de la tastatură a unui cuvânt sau serie de cuvinte. Acesta va fi căutat în compoziția titlurilor cărților.

```
"D:\Atestat Info\12B-Maris-Radu-Ioan-Sistem-de-logare-pentru-o-biblioteca.exe"
Bine ati venit, ExempluDeNume
Ce optiune doriti sa alegeti?
LISTA CARTI.....0
CAUTARE CARTE.....1
CAUTARE AUTOR.....2
CAUTARE EDITURA.....3
SORTARE CRONOLOGICA.....4
LOG OUT.....5
Alege o optiune: 1
Nume carte: Ion
Ion.....Liviu Rebreanu.....editura 1.....2000
Au fost gasite 1 rezultate
Apasa Enter...
```

Fig.11

În cazul în care nu există nici un titlu se afișează faptul ca există 0 rezultate.

```
Nume autor: titlu
Au fost gasite 0 rezultate
Apasa Enter...
```

Fig.12

În cazul în care se introduce, din greșeală, un spațiu în introducerea șirului căutat, se va afișa rezultatul dorit atât timp cât literele din compoziție sunt în ordinea corectă.

```
Nume carte: Io n
Ion.....Liviu Rebreanu.....editura 1.....2000
Au fost gasite 1 rezultate
```

Fig.13

Nu contează dacă literele din compoziția șirului tastat sunt litere mari sau litere mici.

```
Nume carte: I O N
Ion.....Liviu Rebreanu.....editura 1.....2000
Au fost gasite 1 rezultate
```

Fig.14

CAUTARE AUTOR - La apăsarea tastei 2 se permite introducerea de la tastatură a unui cuvânt sau serie de cuvinte. Acesta va fi căutat în compoziția numelor autorilor cărților. Se aplică aceleași reguli precum la CAUTARE CARTE.

```
"D:\Atestat Info\12B-Maris-Radu-Ioan-Sistem-de-logare-pentru-o-biblioteca.exe"
Bine ati venit, ExempluDeNume
Ce optiune doriti sa alegeti?
LISTA CARTI.....0
CAUTARE CARTE.....1
CAUTARE AUTOR.....2
CAUTARE EDITURA.....3
SORTARE CRONOLOGICA.....4
LOG OUT.....5
Alege o optiune: 2
Nume autor: eminescu
Volum de poezii.....Mihai Eminescu.....editura 1.....1980
Au fost gasite 1 rezultate
Apasa Enter...
```

Fig.15

CAUTARE EDITURA - La apăsarea tastei 3 se permite introducerea de la tastatura a unui cuvânt sau serie de cuvinte. Acesta va fi căutat în compoziția numelor editurilor cărților. Se aplică aceleași reguli precum la CAUTARE CARTE.

```
"D:\Atestat Info\12B-Maris-Radu-Ioan-Sistem-de-logare-pentru-o-biblioteca.exe"
Bine ati venit, ExempluDeNume
Ce optiune doriti sa alegeti?
LISTA CARTI.....0
CAUTARE CARTE.....1
CAUTARE AUTOR.....2
CAUTARE EDITURA.....3
SORTARE CRONOLOGICA.....4
LOG OUT.....5
Alege o optiune: 3
Nume Editura: minerva
Moara cu Noroc.....Ioan Slavici.....Minerva.....1881
Au fost gasite 1 rezultate
Apasa Enter...
```

Fig.16

SORTARE CRONOLOGICA - La apăsarea tastei 4 se sortează cărțile în funcție de anul apariției și se afișează pe ecran, de la cea mai veche carte la cea mai nouă.

```

"D:\Atestat Info\12B-Maris-Radu-Ioan-Sistem-de-logare-pentru-o-biblioteca.exe"
CAUTARE CARTE.....1
CAUTARE AUTOR.....2
CAUTARE EDITURA.....3
SORTARE CRONOLOGICA.....4
LOG OUT.....5

Alege o optiune: 4

Alexandru Lapusneanu.....Costache Negruzzi.....Unirea.....1840
Harap-Alb.....Ion Creanga.....Accent.....1877
Moara cu Noroc.....Ioan Slavici.....Minerva.....1881
O scrisoare pierduta.....Ion Luca Caragiale.....Dacia.....1884
Poemele luminii.....Lucian Blaga.....Unirea.....1919
Cuvinte potrivite.....Tudor Arghezi.....Albatros.....1927
Enigma Otiliei.....George Calinescu.....Recif.....1938
Volum de poezii.....Mihai Eminescu.....editura 1.....1980
Amintiri din Copilarie.....Ion Creanga.....editura 2.....2000
Ion.....Liviu Rebreanu.....editura 1.....2000

Apasa Enter...

```

Fig.17

LOG OUT – La apăsarea tastei 5, aplicația se întoarce la meniul de logare.

Meniul Administrator

Administratorul are access la aceleași permisiuni precum utilizatorul „Client”, dar mai are și permisiunea de edita cărțile. Pentru ca deosebirea dintre cele 2 tipuri de utilizatori să fie cât mai clară, meniul are un mesaj de întâmpinare diferit, cât și o paletă de culori diferită.

```
"D:\Atestat Info\12B-Maris-Radu-loan-Sistem-de-logare-pentru-o-biblioteca.exe"
Buna Ziua, Admin
Ce optiune doriti sa alegeti?

LISTA CARTI.....0
CAUTARE CARTE.....1
CAUTARE AUTOR.....2
CAUTARE EDITURA.....3
SORTARE CRONOLOGICA.....4
ADAUGARE CARTE.....5
STERGERE CARTE.....6
LOG OUT.....7

Alege o optiune:
```

Fig.18

ADAUGARE CARTE – Prin apăsarea tastei 5 o nouă înregistrare este introdusă în bibliotecă.

```
"D:\Atestat Info\12B-Maris-Radu-loan-Sistem-de-logare-pentru-o-biblioteca.exe"
Buna Ziua, Admin
Ce optiune doriti sa alegeti?

LISTA CARTI.....0
CAUTARE CARTE.....1
CAUTARE AUTOR.....2
CAUTARE EDITURA.....3
SORTARE CRONOLOGICA.....4
ADAUGARE CARTE.....5
STERGERE CARTE.....6
LOG OUT.....7

Alege o optiune: 5

Titlu: Exemplu De Carte
Autor: Exemplu de Autor
Editura: Exemplu de editura
An Publicare: 2021
Cod ISBN: 1000000000000

Apasa Enter...
```

Fig.19

În bibliotecă pot exista 2 cărți cu același nume, autor, editură și an, însă nu pot exista 2 cărți cu același cod ISBN.

```

D:\Atestat Info\128-Maris-Radu-Ioan-Sistem-de-logare-pentru-o-biblioteca.exe
Buna Ziua, Admin

Ce optiune doriti sa alegeti?

LISTA CARTI.....0
CAUTARE CARTE.....1
CAUTARE AUTOR.....2
CAUTARE EDITURA.....3
SORTARE CRONOLOGICA.....4
ADAUGARE CARTE.....5
STERGERE CARTE.....6
LOG OUT.....7

Alege o optiune: 5

Titlu: Carte Exemplu
Autor: Autor exemplu
Editura: editura exemplu
An Publicare: 1800
Cod ISBN: 1000002154785
Exista in biblioteca o carte cu acest cod ISBN

Moara cu Noroc.....Ioan Slavici.....Minerva.....1881

Apasa Enter...

```

Fig.20

```

Titlu: Moara cu Noroc
Autor: Ioan Slavici
Editura: Minerva
An Publicare: 1881
Cod ISBN: 1000000000000

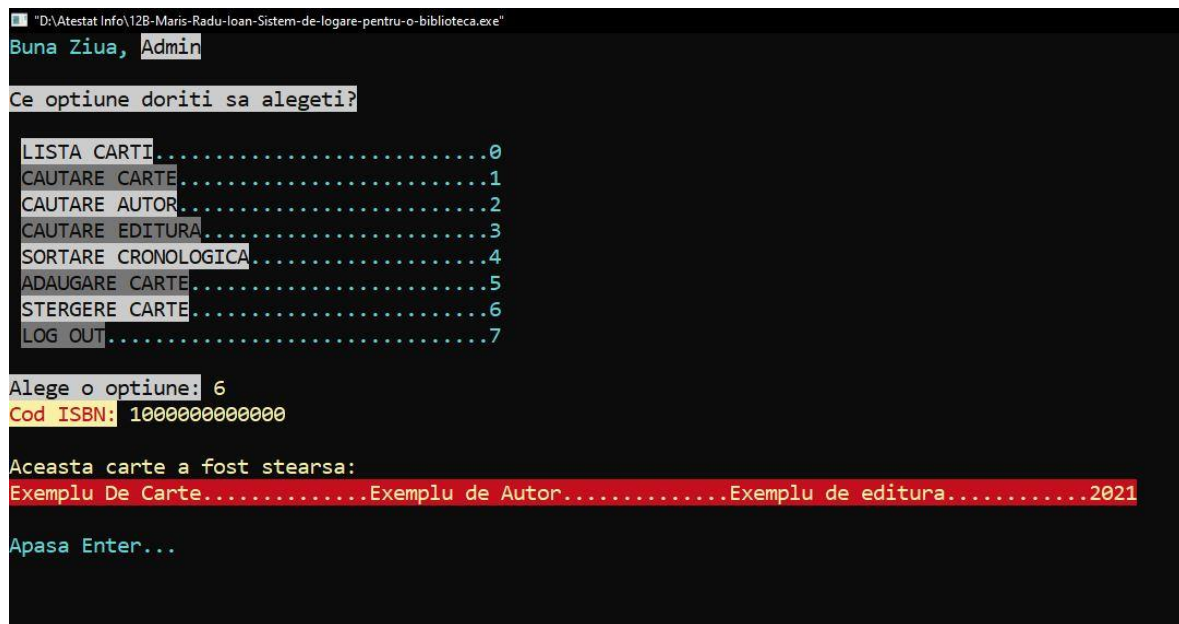
```

Fig.21

Titlu	Autor	Editura	An publicare	Cod ISBN
Alexandru Lapusneanu	Costache Negruzzi	Unirea	1840	9574861235486
Amintiri din Copilarie	Ion Creanga	editura 2	2000	9789739987452
Cuvinte potrivite	Tudor Arghezi	Albatros	1927	9578452136542
Enigma Otiliei	George Calinescu	Recif	1938	4587695845784
Harap-Alb	Ion Creanga	Accent	1877	2154215876589
Ion	Liviu Rebreanu	editura 1	2000	9789732365478
Moara cu Noroc	Ioan Slavici	Minerva	1881	1000002154785
Moara cu Noroc	Ioan Slavici	Minerva	1881	1000000000000
O scrisoare pierduta	Ion Luca Caragiale	Dacia	1884	2548754236548
Poezele luminii	Lucian Blaga	Unirea	1919	4586952365891
Volum de poezii	Mihai Eminescu	editura 1	1980	9781234567890

Fig.22

STERGERE CARTE – prin apăsarea tastei 6, Administratorul citește de la tastatură un cod ISBN, iar cartea care are codul ISBN identic este ștearsă.



```
"D:\Atestat Info\12B-Maris-Radu-loan-Sistem-de-logare-pentru-o-biblioteca.exe"
Buna Ziua, Admin
Ce optiune doriti sa alegeti?
LISTA CARTI.....0
CAUTARE CARTE.....1
CAUTARE AUTOR.....2
CAUTARE EDITURA.....3
SORTARE CRONOLOGICA.....4
ADAugARE CARTE.....5
STERGERE CARTE.....6
LOG OUT.....7
Alege o optiune: 6
Cod ISBN: 100000000000
Aceasta carte a fost stearsa:
Exemplu De Carte.....Exemplu de Autor.....Exemplu de editura.....2021
Apasa Enter...
```

Fig.22

Dacă nu sunt cărți în bibliotecă cu cod identic cu cel introdus, se afișează un mesaj sugestiv.



```
"D:\Atestat Info\12B-Maris-Radu-loan-Sistem-de-logare-pentru-o-biblioteca.exe"
Buna Ziua, Admin
Ce optiune doriti sa alegeti?
LISTA CARTI.....0
CAUTARE CARTE.....1
CAUTARE AUTOR.....2
CAUTARE EDITURA.....3
SORTARE CRONOLOGICA.....4
ADAugARE CARTE.....5
STERGERE CARTE.....6
LOG OUT.....7
Alege o optiune: 6
Cod ISBN: 0
Nu exista carte cu codul ISBN introdus
Apasa Enter...
```

Fig.23

Cod Sursă

```
#include <iostream>
#include <fstream>
#include <conio.h>
#include <windows.h>
#include <iomanip>
#include <cstring>
using namespace std;

struct cont{
    char username[200],parola[200];
};

//definirea sturcturii cont care sa retină numele de utilizator si parola

struct biblioteca{
    char titlu[200],autor[200],editura[200];
    int anpublic;
    long long codISBN;
};

/*
definirea structurii biblioteca care retine informații despre o carte: titlu, autor, editura, anul publicării si codul
ISBN
*/

int k=0,nr_persoane,nr_carti,n1,n2;
char c,a[200],b[200];
cont utilizator[200];
biblioteca carte[200],carte2[200],aux;

HANDLE color=GetStdHandle(STD_OUTPUT_HANDLE); //permite folosirea de culori in consolă

void radiere_spatii(char s[200]){
    char s1[200]="";
    char *p;
    int ok=0;
    p=strtok(s," ");
    while(p!=NULL){
        strcat(s1,p);
        p=strtok(NULL," ");
        ok++;
    }
    s[strlen(s1)]='\0';
    for(int k=0;k<strlen(s1);k++)
        s[k]=s1[k];
    return;
}

/*
subprogramul radiere_spatii primește un șir de caractere si șterge toate spațiile, mai exact toate caracterele '
' din intermediul acestuia
*/

void litere_mici(char s[200]){
    for(int i=0;i<strlen(s);i++){
        if(s[i]>='A' && s[i]<='Z')
            s[i]=s[i]+'a'-'A';
    }
}
```



```

    return;
}

```

//subprogramul litere_mici primește în sir de caractere și transformă toate literele mari în litere mici

```

void citire(cont utilizator[200],int &nr_persoane){
    ifstream f1("Utilizatori.txt");
    int k=0;
    while(!f1.eof()){
        k++;
        f1.getline(utilizator[k].username,22);
        f1.getline(utilizator[k].parola,22);
    }
    nr_persoane=k;
    f1.close();
    return;
}

```

//citeste utilizatorii din fisierul „Utilizatori” pentru a folosi informatiile în decursul programului

```

void adugare_ut(int &nr_persoane,cont utilizator[200]){ //adaugare_ut permite adăugarea unui utilizator în
    for(int i=1;i<=75;i++) // baza de date cu nume de utilizatori și parole
        cout<<"~"; //element de design
    cout<<endl;
    nr_persoane++; //marește numărul de persoane din baza de date
    SetConsoleTextAttribute(color,14);
    cout<<"Introduce-ti numele și parola pe care doriți să le aveți"<<endl<<endl;
    SetConsoleTextAttribute(color,228);
    cout<<"Username:";
    SetConsoleTextAttribute(color,14);
    cout<<" "; //elemente de design
    cin>>utilizator[nr_persoane].username; //citirea de la tastatura a numelui de utilizator
    SetConsoleTextAttribute(color,228);
    cout<<endl<<"Parola:";
    SetConsoleTextAttribute(color,14);
    cout<<" "; //elemente de design
    cin>>utilizator[nr_persoane].parola; //citirea de la tastatura a parolei
    for(int i=1;i<nr_persoane;i++)
        if(strcmp(utilizator[nr_persoane].username,utilizator[i].username)==0){
            SetConsoleTextAttribute(color,78);
            cout<<endl<<"Va rugăm alegeți alt nume deoarece acesta este folosit de un alt utilizator"<<endl;
            SetConsoleTextAttribute(color,14);
            nr_persoane--; //verifică toți utilizatorii, iar în cazul în care există deja numele de utilizator
            for(int i=1;i<=75;i++) // în baza de date, se afișează un mesaj sugestiv, iar ultima înregistrare este
                cout<<"~"; //ștersă
            cout<<endl;
            return; //programul nu mai continuă executarea, ci se oprește
        }
    cout<<endl<<"Contul a fost creat cu succes!"<<endl; //în cazul în care nu a mai fost folosit numele
    for(int i=1;i<=75;i++)
        cout<<"~";
    cout<<endl;
    return;
}

```

```

void citire_carti(biblioteca carte[200],int &nr_carti){
    ifstream f2("Carti.txt");
    f2>>nr_carti;
    f2.get();
    for(int k=1;k<=nr_carti;k++){
        f2.getline(carte[k].titlu,200);
    }
}

```

```

        f2.getline(carte[k].autor,200);
        f2.getline(carte[k].editura,200);
        f2>>carte[k].anpublic;
        f2>>carte[k].codISBN;
        f2.get();
    }
    f2.close();
    return;
}

```

//subprogramul citire_carti citeste inregistrarile din fisierul „Carti.txt” in vectorul carte[200]

```

void afisare_generica(biblioteca carte[200],int nr_carti,int k,int ok){
    int i,spatiere=30;                                //spatiere are rolul de a stabili distanța dintre coloanele
    SetConsoleTextAttribute(color,ok);                // titlu, autor, editura și cod ISBN
    cout<<endl<<carte[k].titlu;
    for(i=1;i<=spatiere-strlen(carte[k].titlu);i++)    // afișează o linie de puncte inte 2 coloane, incepand de la
        cout<<" ";                                    // sfarsitul cuvantului din prima coloana până la inceputul
    cout<<carte[k].autor;                              // celei de a 2-a coloane, repetându-se pentru fiecare grup
    for(i=1;i<=spatiere-strlen(carte[k].autor);i++)    // de 2 coloane: (titlu; autor), (autor; editura), (editura; an).
        cout<<" ";
    cout<<carte[k].editura;
    for(i=1;i<=spatiere-strlen(carte[k].editura);i++)
        cout<<" ";
    cout<<carte[k].anpublic;
    SetConsoleTextAttribute(color,14);
    cout<<endl;
    return;
}

```

/*
subprogramul afisare_generica primeste lista de carti, numărul de cărți din biblioteca si o valoare intreaga
care are rolul de a stabili culoarea scrisului
*/

```

void afisare_carti(biblioteca carte[200], int nr_carti){
    int i=1,spatiere=25;
    system("CLS");
    SetConsoleTextAttribute(color,176);
    cout<<"Titlu";
    for(i=1;i<=spatiere-strlen("Titlu")+1;i++)
        cout<<" ";
    cout<<"Autor";
    for(i=1;i<=spatiere-strlen("Autor")+1;i++)
        cout<<" ";
    cout<<"Editura";
    for(i=1;i<=spatiere-strlen("Editura")+1;i++)
        cout<<" ";
    cout<<"An publicare";
    for(i=1;i<=spatiere-strlen("An punlicare")+1;i++)
        cout<<" ";
    cout<<"Cod ISBN    "<<endl;
    SetConsoleTextAttribute(color,11);
    for(i=1;i<=117;i++){
        cout<<"~";
    }
    cout<<endl;
    SetConsoleTextAttribute(color,14);
    int ok=0;
    for(int k=1;k<=nr_carti;k++){

```

```

    if(ok==0){
        SetConsoleTextAttribute(color,11);
        ok=1;
    }
    else{
        SetConsoleTextAttribute(color,176);
        ok=0;
    }
    cout<<endl<<carte[k].titlu;
    for(i=1;i<=spatiere-strlen(carte[k].titlu);i++)
        cout<<" ";
    cout<<"|";
    cout<<carte[k].autor;
    for(i=1;i<=spatiere-strlen(carte[k].autor);i++)
        cout<<" ";
    cout<<"|";
    cout<<carte[k].editura;
    for(i=1;i<=spatiere-strlen(carte[k].editura);i++)
        cout<<" ";
    cout<<"|";
    cout<<carte[k].anpublic;
    for(i=1;i<=spatiere-4;i++)
        cout<<" ";
    cout<<"|";
    cout<<carte[k].codISBN;
}
SetConsoleTextAttribute(color,11);
return;
}

```

//subprogramul afisare carti are rolul de a afisa lista completa de informatii privind cartiile din biblioteca

```

void afisare_cronologica(biblioteca carte[200],int nr_carti){
    biblioteca aux; //aux e de tip biblioteca, structura definita la inceputul programului
    int ok=0,i;
    do{
        ok=0;
        for(int k=1;k<nr_carti;k++){
            if(carte[k].anpublic>carte[k+1].anpublic){ //se compara anii
                aux=carte[k]; //metoda paharelor pentru a schimba 2 valori intre ele
                carte[k]=carte[k+1];
                carte[k+1]=aux;
                ok=1;
            }
        }
    }while(ok!=0); //sortare prin metoda metoda bulelor (bubble sort)
    for(k=1;k<=nr_carti;k++)
        afisare_generica(carte,nr_carti,k,144);
    return;
}

```

// subprogramul ordoneaza cartile in functie de anul aparitiei si afiseaza toate cartiile

```

void sortare_alfabetica(biblioteca carte[200],int nr_carti){
    biblioteca aux;
    for(int i=1;i<nr_carti;i++)
        for(int j=i;j<=nr_carti;j++){
            if(strcmp(carte[i].titlu,carte[j].titlu)==1){ //compara 2 şiruri de caractere
                aux=carte[i]; //metoda interschimbării
                carte[i]=carte[j];
                carte[j]=aux;
            }
        }
    }

```

```

return;
}

```

//sortare_alfabetica ordoneaza cartiile in functie de titlu in ordine alfabetica

```

void adaugare_carte(biblioteca carte[200], biblioteca carte2[200], int &nr_carti){
    nr_carti++;
    cin.get();
    SetConsoleTextAttribute(color, 228);
    cout<<endl<<"Titlu:";
    SetConsoleTextAttribute(color, 14);
    cout<<" ";
    cin.getline(carte[nr_carti].titlu, 200);
    SetConsoleTextAttribute(color, 228);
    cout<<endl<<"Autor:";
    SetConsoleTextAttribute(color, 14);
    cout<<" ";
    cin.getline(carte[nr_carti].autor, 200);
    SetConsoleTextAttribute(color, 228);
    cout<<endl<<"Editura:";
    SetConsoleTextAttribute(color, 14);
    cout<<" ";
    cin.getline(carte[nr_carti].editura, 200);
    SetConsoleTextAttribute(color, 228);
    cout<<endl<<"An Publicare:";
    SetConsoleTextAttribute(color, 14);
    cout<<" ";
    cin>>carte[nr_carti].anpublic;
    cin.get();
    SetConsoleTextAttribute(color, 228);
    cout<<endl<<"Cod ISBN:";
    SetConsoleTextAttribute(color, 14);
    cout<<" ";
    cin>>carte[nr_carti].codISBN;
    int i=1; //o carte nouă este adugata in baza de date
    while(i<nr_carti){
        if(carte[i].codISBN==carte[nr_carti].codISBN){
            nr_carti--;
            SetConsoleTextAttribute(color, 228);
            cout<<"Exista in biblioteca o carte cu acest cod ISBN"<<endl;
            afisare_generica(carte, nr_carti, i, 206);
            return;
        }
        i++;
    } //in cazul în care există deja o carte cu acelasi cod ISBN, cartea recent introdusa va fi ştearsă
    carte2[nr_carti]=carte[nr_carti]; //in paralel se adauga in vectorul carte2
    radiere_spatii(carte2[nr_carti].titlu); //in vectorul carte2 toate spatiile sunt sterse
    radiere_spatii(carte2[nr_carti].autor);
    radiere_spatii(carte2[nr_carti].editura);
    litere_mici(carte2[nr_carti].titlu); //in vectorul carte2 toate literele devin litere mici
    litere_mici(carte2[nr_carti].autor);
    litere_mici(carte2[nr_carti].editura);
    cin.get();
    return;
}

```

```

void cautare_titlu(biblioteca carte2[200], biblioteca carte[200], int nr_carti){
    SetConsoleTextAttribute(color, 228);
    cout<<"\nNume carte:";
    SetConsoleTextAttribute(color, 14);

```

```

cout<<" ";
char p[100];
cin.get();
cin.getline(p,100);
litere_mici(p);
radiere_spatii(p);           //transformă cuvântul căutat în litere mici și șterge spațiile
int gasit=0;
for(int k=1;k<=nr_carti;k++){
    if(strstr(carte2[k].titlu,p)!=NULL){           //caută șirul transformat în carte2, vectorul cu toate spațiile
        gasit++;                                 //șterse și toate literele mici și afișează corespondentul în
        afisare_generica(carte,nr_carti,k,32);    //vectorul carte, versiunea needitată a înregistrărilor
    }
}
cout<<endl<<"Au fost gasite ";
SetConsoleTextAttribute(color,228);
cout<<gasit;
SetConsoleTextAttribute(color,14);
cout<<" rezultate"<<endl;           //afișează numărul de rezultate gasite
return;
}

```

//strstr caută un într-un șir un subsir
 //în cazul acesta, se citește de la tastatură un șir de caractere, se șterg spațiile și toate literele și caută
 //subsșirul tastat în șirul titlu aparținând vectorului carte2, iar mai apoi afișează corespondentul din carte1

```

void cautare_autor(biblioteca carte2[200],biblioteca carte[200],int nr_carti){
    SetConsoleTextAttribute(color,228);
    cout<<"\nNume autor:";
    SetConsoleTextAttribute(color,14);
    cout<<" ";
    char p[100];
    cin.get();
    cin.getline(p,100);
    radiere_spatii(p);
    litere_mici(p);
    int gasit=0;
    for(int k=1;k<=nr_carti;k++){
        if(strstr(carte2[k].autor,p)!=NULL){
            gasit++;
            afisare_generica(carte,nr_carti,k,160);
        }
    }
    cout<<endl<<"Au fost gasite ";
    SetConsoleTextAttribute(color,228);
    cout<<gasit;
    SetConsoleTextAttribute(color,14);
    cout<<" rezultate"<<endl;
    return;
}

```

//Același program cu cautare_titlu, însă șirul citit de la tastatură este căutat în șirul autor din vectorul carte2

```

void cautare_editura(biblioteca carte2[200],biblioteca carte[200],int nr_carti){
    SetConsoleTextAttribute(color,228);
    cout<<"\nNume Editura:";
    SetConsoleTextAttribute(color,14);
    cout<<" ";
    char p[100];
    cin.get();
    cin.getline(p,100);
    radiere_spatii(p);
}

```

```

    litere_mici(p);
    int gasit=0;
    for(int k=1;k<=nr_carti;k++){
        if(strstr(carte2[k].editura,p)!=NULL){
            gasit++;
            afisare_generica(carte,nr_carti,k,95);
        }
    }
    cout<<endl<<"Au fost gasite ";
    SetConsoleTextAttribute(color,228);
    cout<<gasit;
    SetConsoleTextAttribute(color,14);
    cout<<" rezultate"<<endl;
    return;
}

void design_titlu(){
    SetConsoleTextAttribute(color,228);
    for(int i=1;i<=74;i++)
        cout<<" ";
    cout<<endl;
    cout<<" ";
    SetConsoleTextAttribute(color,78);
    cout<<"      @      @      ";
    SetConsoleTextAttribute(color,228);
    cout<<" "<<endl<<" ";
    SetConsoleTextAttribute(color,78);
    cout<<" ##### # ##### # # ##### ##### # ##### # # # ";
    SetConsoleTextAttribute(color,228);
    cout<<" "<<endl<<" ";
    SetConsoleTextAttribute(color,78);
    cout<<" # # # # # # # # # # # # # # # # # ";
    SetConsoleTextAttribute(color,228);
    cout<<" "<<endl<<" ";
    SetConsoleTextAttribute(color,78);
    cout<<" ##### # ##### # # # # # # ##### # ##### ";
    SetConsoleTextAttribute(color,228);
    cout<<" "<<endl<<" ";
    SetConsoleTextAttribute(color,78);
    cout<<" # # # # # # # # # # # # # # # # # ";
    SetConsoleTextAttribute(color,228);
    cout<<" "<<endl<<" ";
    SetConsoleTextAttribute(color,78);
    cout<<" ##### # ##### ##### # ##### # ##### # # # ";
    SetConsoleTextAttribute(color,228);
    cout<<" "<<endl<<" ";
    SetConsoleTextAttribute(color,78);
    for(int i=1;i<=72;i++)
        cout<<" ";
    SetConsoleTextAttribute(color,228);
    cout<<" "<<endl;
    SetConsoleTextAttribute(color,228);
    for(int i=1;i<=74;i++)
        cout<<" ";
    cout<<endl<<endl;
    return;
}

```

//subprogramul design_titlu are rol de a creea o experiență de utilizare plăcută

```
void meniu_principal(char y[200],int z,int k,int ok,int ok1,int ok2){
```

```

SetConsoleTextAttribute(color,11);
for(int i=1;i<=ok1;i++)
    cout<<" ";
SetConsoleTextAttribute(color,ok);
cout<<y;
SetConsoleTextAttribute(color,ok2);
for(int i=1;i<=z-strlen(y)-1;i++)
    cout<<" ";
cout<<k<<endl;
return;
}

```

//z reprezinta distanța ideală dintre sirul de caractere si numărul asociat acestuia
//k reprezinta un număr afișat pentru design, astfel utilizatorul asociând șirului de caractere un număr
//ok reprezintă culoarea pe care o are șirul de caractere y la afișarea în consolă
//ok1 reprezintă spațierea la stânga a sirului de caractere y
//ok2 reprezintă culoarea pe care spatiul dintre sirul de caractere si numărul afișat

```

void scriere_in_fisier_carti(biblioteca carte[200],int nr_carti){
    ofstream g2("Carti.txt");
    g2<<nr_carti;
    for(k=1;k<=nr_carti;k++){
        g2<<endl<<carte[k].titlu;
        g2<<endl<<carte[k].autor;
        g2<<endl<<carte[k].editura;
        g2<<endl<<carte[k].anpublic;
        g2<<endl<<carte[k].codISBN;
    }
    g2.close();
    return;
}

```

//scriere in fișierul Carti.txt a informațiilor legate de cărți

```

void stergere_carte(biblioteca carte[200],biblioteca carte2[200],int &nr_carti){
    long long z;
    int i=1;
    SetConsoleTextAttribute(color,228);
    cout<<"Cod ISBN:";
    SetConsoleTextAttribute(color,14);
    cout<<" ";
    cin>>z;
    while(z!=carte[i].codISBN && i<=nr_carti){
        i++;
    }
    if(i==nr_carti+1){
        SetConsoleTextAttribute(color,78);
        cout<<endl<<"Nu exista carte cu codul ISBN introdus";
        SetConsoleTextAttribute(color,14);
        cout<<endl;
    }
    else{
        cout<<endl<<"Aceasta carte a fost stearsa:";
        afisare_generica(carte,nr_carti,i,78);
        while(i<nr_carti){
            carte[i]=carte[i+1];
            carte2[i]=carte2[i+1];
            i++;
        }
        nr_carti--;
    }
}

```

```

    return;
}

void ajutor(){
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);
    cout<<"\n-----Ajutor-----\n";
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),228);
    ifstream f("Help.txt");
    char rand[256];
    while(f.getline(rand,256))
        cout<<rand<<"\n";
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);
    cout<<"-----";
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);
    f.close();
}

```

// subprogramul afișează conținutul fișierului „Help.txt” menit să ajute la înțelegerea aplicației

```

int main(){
    citire(utilizator,nr_persoane);
    citire_carti(carte,nr_carti);
    do{
        for(int i=1;i<=nr_carti;i++){
            carte2[i]=carte[i];
            radiere_spatii(carte2[i].titlu);
            radiere_spatii(carte2[i].autor);
            radiere_spatii(carte2[i].editura);
        }
        for(int i=1;i<=nr_carti;i++){
            litere_mici(carte2[i].titlu);
            litere_mici(carte2[i].autor);
            litere_mici(carte2[i].editura);
        }
        //transforma toate șirurile de caractere în șiruri fără spații și litere mici
        design_titlu();

        SetConsoleTextAttribute(color,14);
        char x[200]="APLICATIE PERMIS DE BIBLIOTECA";
        int z = strlen(x);
        for(int i=1;i<=36-z/2;i++)
            cout<<" ";
        SetConsoleTextAttribute(color,228);
        cout<<x<<endl<<endl;

        meniu_principal("ACCESARE CONT",z,0,78,36-z/2,14);
        meniu_principal("CREEARE CONT",z,1,78,36-z/2,14);
        meniu_principal("SCHIMBARE PAROLA",z,2,78,36-z/2,14);
        meniu_principal("AJUTOR",z,3,78,36-z/2,14);
        meniu_principal("EXIT",z,4,78,36-z/2,14);

        SetConsoleTextAttribute(color,14);
        cout<<endl<<"Alege o optiune: ";
        cin>>n1;
        cout<<endl;
        //switch are rolul de a verifica dacă valoarea din interiorul parantezelor rotunde este
        //unul din cazurile specificate, iar dacă da, codul din interiorul cazului se execută
        switch(n1){
            case 0:{
                char a[200],b[200];
                SetConsoleTextAttribute(color,228);
                cout<<"Username:";
                SetConsoleTextAttribute(color,14);
                cout<<" ";
            }
        }
    }while(1);
}

```



```

cin>>a;
SetConsoleTextAttribute(color,228);
cout<<endl<<"Parola:";
SetConsoleTextAttribute(color,14);
cout<<" ";
cin>>b;
int validare_user=0,admin=0;
if(strcmp(a,utilizator[1].username)==0 && strcmp(b,utilizator[1].parola)==0){
    admin=1; //valoarea admin se schimbă atunci când se introduce corect numele de
    system("CLS"); // administrator și parola
    do{
        sortare_alfabetica(carte2,nr_carti);
        for(int i=1;i<=nr_carti;i++)
            for(int j=i;j<=nr_carti;j++)
                if(carte2[i].codISBN==carte[j].codISBN){
                    aux=carte[i];
                    carte[i]=carte[j];
                    carte[j]=aux;
                }
        //programul efectuează o ordonare alfabetică a listei de cărți de fiecare dată când se alege
        //o opțiune din cele posibile acestuia
        SetConsoleTextAttribute(color,11);
        cout<<"Buna Ziua, ";
        SetConsoleTextAttribute(color,112);
        cout<<utilizator[1].username<<endl<<endl;
        cout<<"Ce opțiune doriți să alegeți?"<<endl<<endl;
        SetConsoleTextAttribute(color,14);
        meniu_principal("LISTA CARTI",40,0,112,1,11);
        meniu_principal("CAUTARE CARTE",40,1,128,1,11);
        meniu_principal("CAUTARE AUTOR",40,2,112,1,11);
        meniu_principal("CAUTARE EDITURA",40,3,128,1,11);
        meniu_principal("SORTARE CRONOLOGICA",40,4,112,1,11);
        meniu_principal("ADAUGARE CARTE",40,5,128,1,11);
        meniu_principal("STERGERE CARTE",40,6,112,1,11);
        meniu_principal("LOG OUT",40,7,128,1,11);
        cout<<endl;
        SetConsoleTextAttribute(color,112);
        cout<<"Alege o opțiune:";
        SetConsoleTextAttribute(color,14);
        cout<<" ";
        cin>>n2;
        switch(n2){ //se apelează subprogramele descrise anterior
            case 0:{
                afisare_carti(carte,nr_carti);
                SetConsoleTextAttribute(color,11);
                cout<<"\n\nApasa Enter...";
                c=_getch();
                break;
            }
            case 1:{
                cautare_titlu(carte2,carte,nr_carti);
                SetConsoleTextAttribute(color,11);
                cout<<"\n\nApasa Enter...";
                c=_getch();
                break;
            }
            case 2:{
                cautare_autor(carte2,carte,nr_carti);
                SetConsoleTextAttribute(color,11);
                cout<<"\n\nApasa Enter...";
                c=_getch();
            }
        }
    }
}

```

```

        break;
    }
    case 3:{ //se apelează subprogramele descrise anterior
        cautare_editura(carte2, carte, nr_carti);
        SetConsoleTextAttribute(color, 11);
        cout<<"\nApasa Enter...";
        c=_getch();
        break;
    }
    case 4:{
        afisare_cronologica(carte, nr_carti);
        SetConsoleTextAttribute(color, 11);
        cout<<"\nApasa Enter...";
        c=_getch();
        break;
    }
    case 5:{
        adaugare_carte(carte, carte2, nr_carti);
        scriere_in_fisier_carti(carte, nr_carti);
        SetConsoleTextAttribute(color, 11);
        cout<<"\nApasa Enter...";
        c=_getch();
        break;
    }
    case 6:{
        stergere_carte(carte, carte2, nr_carti);
        scriere_in_fisier_carti(carte, nr_carti);
        SetConsoleTextAttribute(color, 11);
        cout<<"\nApasa Enter...";
        c=_getch();
        break;
    }
    }system("CLS");
}while (n2!=7);
}
else{
    //dacă nu ne aflăm în contul de Admin, iar contul se află în fișierul utilizatori, atunci
    //validare_user își modifică valoarea
    for(int k=1; k<=nr_persoane; k++)
        if(strcmp(a, utilizator[k].username)==0 && strcmp(b, utilizator[k].parola)==0)
            validare_user=k;
    if(validare_user==0){
        SetConsoleTextAttribute(color, 78);
        cout<<"\nNumele contului sau parola sunt gresite!";
        SetConsoleTextAttribute(color, 14);
        cout<<"\nApasa o tasta";
        c=_getch();
    }
    else{
        system("CLS");
        citire_carti(carte, nr_carti);
        do{
            //ordonare alfabetica
            sortare_alfabetica(carte2, nr_carti);
            for(int i=1; i<=nr_carti; i++)
                for(int j=i; j<=nr_carti; j++)
                    if(carte2[i].codISBN==carte[j].codISBN){
                        aux=carte[i];
                        carte[i]=carte[j];
                        carte[j]=aux;
                    }
                }
            }
        }
    }
}

```

```

    }
    //sfarsit ordonare alfabetica
    cout<<"Bine ati venit, ";
    SetConsoleTextAttribute(color,206);
    cout<<utilizator[validare_user].username<<endl<<endl;
    cout<<"Ce optiune doriti sa alegeti?"<<endl<<endl;
    SetConsoleTextAttribute(color,14);
    meniu_principal("LISTA CARTI",40,0,206,1,14);
    meniu_principal("CAUTARE CARTE",40,1,78,1,14);
    meniu_principal("CAUTARE AUTOR",40,2,206,1,14);
    meniu_principal("CAUTARE EDITURA",40,3,78,1,14);
    meniu_principal("SORTARE CRONOLOGICA",40,4,206,1,14);
    meniu_principal("LOG OUT",40,5,78,1,14);
    cout<<endl;
    SetConsoleTextAttribute(color,206);
    cout<<"Alege o optiune:";
    SetConsoleTextAttribute(color,14);
    cout<<" ";
    cin>>n2;
    switch(n2){
        case 0:{
            afisare_carti(carte,nr_carti);
            cout<<"\nApasa Enter...";
            SetConsoleTextAttribute(color,14);
            c=_getch();
            break;
        }
        case 1:{
            cautare_titlu(carte2,carte,nr_carti);
            cout<<"\nApasa Enter...";
            SetConsoleTextAttribute(color,14);
            c=_getch();
            break;
        }
        case 2:{
            cautare_autor(carte2,carte,nr_carti);
            cout<<"\nApasa Enter...";
            SetConsoleTextAttribute(color,14);
            c=_getch();
            break;
        }
        case 3:{
            cautare_editura(carte2,carte,nr_carti);
            cout<<"\nApasa Enter...";
            SetConsoleTextAttribute(color,14);
            c=_getch();
            break;
        }
        case 4:{
            afisare_cronologica(carte,nr_carti);
            cout<<"\nApasa Enter...";
            SetConsoleTextAttribute(color,14);
            c=_getch();
            break;
        }
    }
    }system("CLS");
    }while (n2!=5);
    }
    }
    break;
}

```

```

case 1:{
    adugare_ut(nr_persoane,utilizator);
    ofstream g1("Utilizatori.txt");
    //adminul
    g1<<utilizator[1].username<<endl;
    g1<<utilizator[1].parola;
    //restul utilizatorilor
    for(k=2;k<=nr_persoane;k++){
        g1<<endl<<utilizator[k].username;
        g1<<endl<<utilizator[k].parola;
    }
    cout<<"\nApasati o tasta...";
    c=_getch();
    break;
}
case 2:{
    char a[200],b[200];
    for(int i=1;i<=75;i++)
        cout<<"~";
    cout<<endl;
    SetConsoleTextAttribute(color,228);
    cout<<"Username:";
    SetConsoleTextAttribute(color,14);
    cout<<" ";
    cin>>a;
    SetConsoleTextAttribute(color,228);
    cout<<endl<<"Vechea parola:";
    SetConsoleTextAttribute(color,14);
    cout<<" ";
    cin>>b;
    int validare_user=0;
    for(int k=1;k<=nr_persoane;k++){
        if(strcmp(a,utilizator[k].username)==0 && strcmp(b,utilizator[k].parola)==0)
            validare_user=k;
    }
    if(validare_user==0){
        SetConsoleTextAttribute(color,78);
        cout<<endl<<"Numele contului sau parola sunt gresite!"<<endl;
        SetConsoleTextAttribute(color,14);
    }
    else{
        SetConsoleTextAttribute(color,228);
        cout<<"\nIntroduce-ti noua parola:";
        SetConsoleTextAttribute(color,14);
        cout<<" ";
        cin>>utilizator[validare_user].parola;
    }
    for(int i=1;i<=75;i++)
        cout<<"~";
    cout<<endl;
    cout<<"\nApasati o tasta...";
    c=_getch();
    break;
}
case 3:{
    ajutor();
    cout<<"\nApasati o tasta...";
    c=_getch();
    break;
}
} system("CLS");
}while(n1!=4);

```

```
ofstream g1("Utilizatori.txt");           //scrierea in fişierul de Utilizatori in urma modificărilor
//adminul
g1<<utilizator[1].username<<endl;
g1<<utilizator[1].parola;

//restul utilizatorilor
for(k=2;k<=nr_persoane;k++){
    g1<<endl<<utilizator[k].username;
    g1<<endl<<utilizator[k].parola;
}
g1.close();
return 0;
}
```

Bibliografie. Webografie

1.	[* * *]	* * *	http://vega.unitbv.ro/~cataron/Courses/PCLPII/PCLP2_Laboratorul2.pdf
2.	[* * *]	* * *	https://ro.wikipedia.org/wiki/C%2B%2B
3.	[* * *]	* * *	https://ro.ourladylakes.org/819789-why-use-conio-h-BJQVZJ
4.	[* * *]	* * *	https://en.wikipedia.org/wiki/Windows.h
5.	[* * *]	* * *	https://www.cplusplus.com/reference/iomanip/