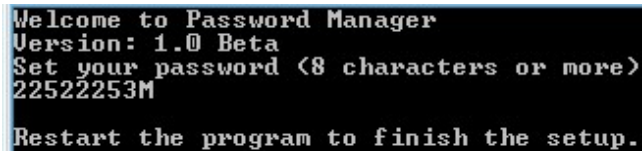


Lab 2

Manager de parole (C++)

Manageru de parole creat are o parolă unică de acces la toată baza de date care este setată la prima execuție a programului, și trebuie să fie mai mare de 7 caractere (fig. 4.1).

Parola master creată este salvată într-un fișier de tip .dat “*Master Password.dat*” utilizându-se cel mai simplu algoritm de criptare *Cesar*.

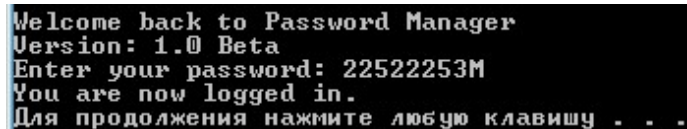


```
Welcome to Password Manager
Version: 1.0 Beta
Set your password <8 characters or more>
22522253M

Restart the program to finish the setup.
-
```

Fig. 4.1

La execuția ulterioară a programului ni se cere să introducem parola master pentru a avea acces la meniul programului (fig. 4.2).



```
Welcome back to Password Manager
Version: 1.0 Beta
Enter your password: 22522253M
You are now logged in.
Для продолжения нажмите любую клавишу . . .
```

Fig. 4.2

După logare pe ecran apare un meniu ce ne oferă opțiunile: de introducere, afișare a datelor, ștergerea bazei de date și generarea unei parole (fig. 4.3).

```
Control Panel
(1) Insert Data
(2) Read Stored Data
(3) Delete Stored Data
(4) Password Generator
(0) Exit
Choice:
```

Fig. 4.3

La tastarea tastei (1) accesăm funcția de introducere a datelor. Aici suntem întrebați câte conturi dorim să creăm apoi oferindu-ne posibilitatea să introducem denumirea contului, numele de utilizator și parola. Datele sunt salvate într-un fișier de tip .txt “*Accounts.txt*” criptate cu cifrul Cezar(fig. 4.4).

```
How many accounts you want to create:2
Account:facebook
Username:Katie
Password:235G555.

The account has been saved.
Account:twitter
Username:Katie
Password:56K.4587
```

Fig. 4.4

La tastarea tastei (2) accesăm funcția de afișare a datelor. Aici sunt decriptate datele din fișierul “*Accounts.txt*” (fig. 4.5).

```
Account  Username  Password
facebook Katie     235G555.
twitter  Katie     56K.4587
Для продолжения нажмите любую клавишу . . . _
```

Fig. 4.5

La tastarea tastei (3) accesăm funcția de ștergere a conținutului fișierului “*Accounts.txt*”(fig. 4.6).

```
All stored data was deleted.
Для продолжения нажмите любую клавишу . . . _
```

Fig. 4.6

La tastarea tastei (4) accesăm funcția de generare a parolei, lungimea căreia este introdusă de la tastatură. Parola este salvată în fișierul “*Password.txt*” (fig. 4.7).

```
How long do you want your password? <7-14>: 8
Your password is: 0!vpr4c0
Для продолжения нажмите любую клавишу . . . _
```

Fig. 4.7

La tastarea tastei (0) ieșim din program.

Funcții utilizate în program

- **string encrypt(string s_input)** – returnează caracterele criptate cu cifrul Cezar, deplasarea 6 poziții
- **string decrypt(string encrypted)** – returnează caracterele decriptate cu cifrul Cezar, deplasarea 6 poziții
- **void encrypt_other(string &s_input)** – returnează caracterele criptate cu cifrul Cezar, deplasarea 4 poziții
- **void decrypt_other(string &encrypted)** – returnează caracterele decriptate cu cifrul Cezar, deplasarea 4 poziții
- **int startProgram()** – funcție de acces la meniu, crează parola master cu ajutorul căruia execută logarea
- **void outputLine(string account, string username, string password)** – formatează afișare datelor, este apelată în funcția *readData()*
- **void insert_id()** – funcție de introducere a datelor, ele sunt salvate în “Accounts.txt”
- **void readData()** – citirea datelor salvate în “Accounts.txt”
- **void cleanData()** – ștergerea “Accounts.txt”
- **void passGenerator()** – funcția de generare a parolei, este salvată în fișierul “Password.txt”
- **void userChoice()** – meniu, apel la funcțiile *insert_id()*, *readData()*, *cleanData()*, *passGenerator()*

Listingul programului *Password Manager.cpp*

```
#include <iostream>
#include <cstring>
#include <fstream>
#include <cstdlib>
#include <iomanip>
#include <ctime>
```

```
using namespace std;
#include "Functions.h"
```

```
int main()
{
    startProgram();
    userChoice();

    cin.ignore(100, '\n');
    cin.get();
    return 0;
}
```

Functions.h

```
const string Version_Num = "1.0 Beta"; //universal version number
string OPEN_PASS;
bool FIRST_OPEN; // variable controls whether or not the user has already opened the
program
int temp, len, i;
string encrypted, decrypted;
string account, username, password;

string encrypt(string s_input)
{
    int temp=s_input.size();
    char input[100];
    for (int a=0;a<=temp;a++)
    {
        input[a]=s_input[a];
```

```

        }
        len = strlen(input);
        for (i=0; i<len; i++)
        {
            input[i] = input[i]-6;
        }
        for (i=0; i<len; i++)
        {
            encrypted += input[i];
        }
        cout << "\n";

        return encrypted;
    }

string decrypt(string encrypted)
{
    int temp=encrypted.size();
    char input[100];
    for (int a=0;a<=temp;a++)
    {
        input[a]=encrypted[a];
    }

    len = strlen(input);
    for (i=0; i<len; i++)
    {
        input[i] = input[i]+6;
    }

    for (i=0; i<len; i++)
    {
        decrypted += input[i];
    }
    cout << "\n";
    return decrypted;
}

void encrypt_other(string &s_input)
{
    int temp=s_input.size();

```

```

char input[100];
for (int a=0;a<=temp;a++)
{
    input[a]=s_input[a];
}

len = strlen(input);
for (i=0; i<len; i++)
{
    input[i] = input[i]-4;
}
s_input = "";
for (i=0; i<len; i++)
{
    s_input+= input[i];
}
cout << "\n";
}

```

```

void decrypt_other(string &encrypted)
{
    int temp=encrypted.size();
    char input[100];
    for (int a=0;a<=temp;a++)
    {
        input[a]=encrypted[a];
    }

    len = strlen(input);
    for (i=0; i<len; i++)
    {
        input[i] = input[i]+4;
    }
    encrypted = "";
    for (i=0; i<len; i++)
    {
        encrypted += input[i];
    }
    cout << "\n";
}

```

```

int startProgram()
{
    // ifstream constructor opens file

```

```

ifstream outMasterPassword("Master Password.dat", ios::in);

if (!outMasterPassword)
{
FIRST_OPEN = true;
}
if (FIRST_OPEN == true)
{
    cout << "Welcome to Password Manager" << "\n"
        << "Version: " << Version_Num << "\n"
        << "Set your password (8 characters or more)" << endl;

    cin >> OPEN_PASS; //sets your password

    while (OPEN_PASS.length() < 7)
    {
        cout << "Your Password must be 8 or more characters. Try again!"
<< endl;

        cin >> OPEN_PASS;
    }

    // ofstream constructor opens file
    ofstream outMasterPassword("Master Password.dat", ios::out);

    // exit program if unable to create file
    if (!outMasterPassword)
    {
        cerr << "We are very sorry but the password file cannot be
found/written to :(\nMake sure the current volume your running this from supports file
writing\n" << endl;

        exit(1);
    }

    encrypt(OPEN_PASS);
    outMasterPassword << encrypted << endl;
    cout << "Restart the program to finish the setup." << endl;
}
else if (FIRST_OPEN == false)
{
    ifstream outMasterPassword("Master Password.dat", ios::in);
    if (!outMasterPassword)
    {
        cerr << "File could not be opened" << endl;
    }
}

```

```

        exit(1);
    }
    outMasterPassword >> encrypted;
    decrypt(encrypted);

    string temp_input;

    cout << "Welcome back to Password Manager" << "\n"
        << "Version: " << Version_Num << "\n"
        << "Enter your password: " ;

    cin >> temp_input;
    while (temp_input != decrypted) //compares user input with the saved
password
    {
        cout<<"I'm sorry but the password was incorrect."<<endl;
        cout<<"Enter your password again."<<endl;
        cin >> temp_input;
    }
    cout<<"You are now logged in.\n";
    system("pause");
}

void outputLine(string account, string username, string password)
{
    cout << left << setw(10) << account << setw(10) << username << setw(13) <<
password ;
}

void insert_id()
{
    // ofstream constructor opens file
    ofstream outAccountFile ("Accounts.txt", ios::out | ios::app);

    // exit program if unable to create file
    if (!outAccountFile)
    {
        cerr << "File could not be opened" << endl;
        exit(1);
    }
    int a;

    cout << "\nHow many accounts you want to create:";

```

```

        cin>>a;
        cout<<endl;

        for(int i=0;i<a; i++)
        {
            cout<<"Account:"; cin>>account;
            cout<<"Username:"; cin>>username;
            cout<<"Password:"; cin>>password;

            // read username and password from cin, then place in file
            encrypt_other(account);
            encrypt_other(username);
            encrypt_other(password);
            //encrypts text before its written to file
            outAccountFile << account <<" " << username <<" " << password <<endl;
            cout << "The account has been saved.\n" << endl;
        }
    }

void readData()
{
    // ifstream constructor opens the file
    ifstream inAccountsFile("Accounts.txt", ios::in);

    // exit program if ifstream could not open file
    if (!inAccountsFile)
    {
        cerr << "File could not be opened" << endl;
        exit(1);
    }

    cout << left << setw(10) << "Account " << setw(10) << "Username" << setw(13) <<
    "Password" << fixed << showpoint;

    // display each account in file
    while (inAccountsFile >> account >> username >> password)
    {
        decrypt_other(account);
        decrypt_other(username);
        decrypt_other(password);
        outputLine(account, username, password);
    }
    cout<<endl<<endl;
    system("pause");
}

```

```

    }

void cleanData()
{
    cout << "\n\t All stored data was deleted.\n" << endl;
    remove("Accounts.txt" );
    system("pause");
}

void passGenerator()
{
    ifstream Pass;
    ofstream temp;
    Pass.open("Password.txt");
    temp.open("temp.txt");

    int length;
    string pass =
        "0123456789"
        "!@#$%^&*"
        "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        "abcdefghijklmnopqrstuvwxyz";
    string password;
    srand(time(NULL));
    cout<<" How long do you want your passowrd? (7-14): ";
    cin>>length;
    cout<<endl;
    cout<<" Your password is: ";
    temp<< "Password ";

    for(int i=0; i<length; i++)
    {
        password[i] = pass[rand() % 70];
        cout<<password[i];

        temp<<password[i];
    }
    cout<<endl;

    Pass.close();
    temp.close();
    remove("Password.txt");
    rename("temp.txt", "Password.txt");
}

```

```

        system("pause");
    }

void userChoice() //menu
{
    if (FIRST_OPEN == false)
    {
        int choice;

        while(choice){
            system("cls");
            cout << "\t\t Control Panel" << endl;
            cout << "\t (1) Insert Data " << endl;
            cout << "\t (2) Read Stored Data" << endl;
            cout << "\t (3) Delete Stored Data" << endl;
            cout << "\t (4) Password Generator" << endl;
            cout << "\t (0) Exit" << endl;
            cout << "Choice:";
            cin >> choice;
            system("cls");

            switch(choice){
                case 1: insert_id();break;
                case 2: readData(); break;
                case 3: cleanData();break;
                case 4: passGenerator();break;
                case 0: exit(1);
                default: cout<<"\a Enter your choice again.\n\n";
                break;
            }
        }
    }
}

```