#include <iostream>

#include <string.h>

#include <string>

#include <Windows.h>

#include <iomanip>

#include <conio.h>

using namespace std;

// String

//int main()

//{

// string text = "";

// text += "Apple";

// text += "Apple";

// text += "Apple";

// cout << text << endl;

// cout << text.size() << endl;

// cout << text.length() << endl;

// cout << text.capacity() << endl;

// text += "Apple";

// cout << text << endl;

// cout << text.size() << endl;

// cout << text.length() << endl;

// cout << text.capacity() << endl;

//

// cout << text.max\_size() << endl;

//

// return 0;

//}

//int main()

//{

// string s(50, '\*');

// cout << s << endl;

// cout << s.capacity() << endl;

// s.resize(10);

// s.shrink\_to\_fit();

// cout << s << endl;

// cout << s.capacity() << endl;

//

// return 0;

//}

//int main()

//{

// string text = "Apple";

// cout << text.capacity() << endl;

// text.reserve(1000);

// cout << text.capacity() << endl;

//

// return 0;

//}

//int main()

//{

// string text = "Happy programmers day";

// cout << text << endl;

// text.clear();

//

// if (text.empty())

// {

// cout << "No Text" << endl;

// }

// else

// {

// cout << text << endl;

// }

//

// return 0;

//}

//

//int main()

//{

// string text = "Hi all";

// cout << text << endl;

// text += " , bye bye";

// cout << text << endl;

// text.append("Salam millet");

// cout << text << endl;

//

// return 0;

//}

//int main()

//{

// string name = "Elvin";

// string surname = "Camalzade";

//

// cout << name + " " + surname << endl;

//

// return 0;

//}

//int main()

//{

// string text = "Happy";

//

// cout << text[0] << endl;

// cout << text.at(0) << endl;

//

// text[0] = 'Z';

// cout << text << endl;

//

// for (int x = 0; x < text.size(); x++)

// {

// cout << text[x] << endl;

//

// }

//

// return 0;

//}

/////////////

//

//int main()

//{

// string text = "Happy";

// cout << text.front() << endl; // ilk elementin referansini qaytarir

// text.front() = 'P';

// cout << text.front() << endl;

//

// cout << text.back() << endl; // son elementin referansini qaytarir

// text.back() = 'i';

// cout << text.back() << endl;

// cout << text << endl;

//

// return 0;

//}

///////////

//

//int main()

//{

// string text = "Happy programmers day";

// cout << text.substr(6,10) << endl;

// cout << text << endl;

//

// return 0;

//}

///////////

//int main()

//{

//insert ile - iteratorla(agilli class)isleyir - her nov container-le nece islemek lazim oldugunu bilir (ilk elementin adresi)

//string text = "Happy programmers day";

/\*cout << text << endl;

text.insert(text.begin(), 'E');

cout << text << endl;

text.insert(text.begin() + 5, 'E');

cout << text << endl;

text.insert(text.end() - 1, 'A');

cout << text << endl;\*/

//return 0;

//}

///////////

//int main()

//{

// string text = "SALAMSALAMSALAMSALAM";

// text.replace(3, 0, "HELLO"); // index x - den yazir // y dene silir

// cout << text << endl;

//

// return 0;

//}

//

//int main()

//{

// string text = "SALAM SALAM";

//

// text.pop\_back();

// cout << text << endl;

// text.push\_back();

// cout << text << endl;

//

// return 0;

//}

//int main()

//{

// string text1 = "salam";

// string text2 = "Salam";

//

// cout << boolalpha << (text1 == text2) << endl; // bir textin digerine beraber olmasini, eyni olmasi yoxlayir // Ideal yoxlama budur

// cout << boolalpha << text1.compare(text2) << endl; // ascii xarakterlerini cemine gore yoxlayir // 0 - dirsa sozler beraberdir, eynidir

//

// return 0;

//}

//int main()

//{

// string text;

// text = "Salam";

// cout << text << endl;

//

// string text2;

// text2.assign("Salam");

// cout << text2 << endl;

//

// // text = smth ve .assign() eynidir

//

// return 0;

//}

//

//int main()

//{

// string text = "Happy programmers day";

// cout << text.find('j') << endl; // Olmayanda ekrana boyuk reqem qaytarir

// cout << text.find('H') << endl; // Olanda ekrana ilk elementin indexini tapir

//

// cout << text.rfind('j') << endl; // .rfind() stringin sagindan isleyir

// cout << text.rfind('y') << endl;

//

// return 0;

//}

//int main()

//{

// string text = "";

// cout << "Enter text : ";

// cin >> text;

// cout << text << endl;

// // cin stringin ilk bosluguna qeder - ilk qirilma noqtesine qeder - goturur

//

// // #include <string>

// cout << "Enter text : ";

// getline(cin, text);

// cout << text << endl;

// return 0;

//}

// Task

//

//bool IsGmail(const string text)

//{

// string newtext\_com = text.substr(text.length() - 10, text.size());

// string newtext\_ru = text.substr(text.length() - 8, text.size());

//

// if ("@gmail.com" == newtext\_com)

// {

// cout << "@gmail.com" << endl;

// return true;

// }

// else if ("@mail.ru" == newtext\_ru)

// {

// cout << "@mail.ru" << endl;

// return true;

// }

//

// return false;

//}

//

//int main()

//{

// string text = "";

// cout << " Enter your gmail : ";

// getline(cin, text);

//

// if (IsGmail(text))

// {

// cout << " Is Correct Mail" << endl;

// }

// else

// {

// cout << " Is Not Correct Mail" << endl;

// }

//

// return 0;

//}

// Task

class User

{

private:

string fullname;

string username;

string password;

public:

User() : fullname(""), username(""), password("") {};

User(const string& fullname, const string& username, const string& password)

{

SetFullname(fullname);

SetUsername(username);

SetPassword(password);

}

string GetUsername() const

{

return username;

}

string GetFullname() const

{

return fullname;

}

string GetPassword() const

{

return password;

}

void SetFullname(const string& fullname)

{

if (!fullname.empty())

{

this->fullname = fullname;

}

}

void SetUsername(const string& username)

{

if (!username.empty())

{

this->username = username;

}

}

void SetPassword(const string& password)

{

if (!password.empty() && password.size() >= 5)

{

this->password = password;

}

}

void Show() const

{

cout << " ======= I N F O =======" << endl;

cout << " Fullname : " << GetFullname() << endl;

cout << " Username : " << GetUsername() << endl;

string password\_s(password.size(), '\*');

cout << " Password : " << password\_s << endl;

}

};

class Database

{

private:

User\*\* users;

int user\_count;

public:

void SetPassword(string& password)

{

while (true)

{

char a = \_getch();

if (a == '\r') {

break;

}

else if (a == 8) {

cout << "\b \b";

password.pop\_back();

}

else {

cout << "\*";

password.push\_back(a);

}

}

}

User GetNewUser()

{

cin.ignore();

cin.clear();

system("cls");

string fullname;

cout << " Enter your fullname : ";

getline(cin, fullname);

string username;

cout << " Enter your username : ";

getline(cin, username);

string password;

cout << " Enter your password : ";

SetPassword(password);

return User{ fullname,username,password };

}

void AddUserToDatabase(const User& newuser)

{

User\*\* newusers = new User\*[user\_count + 1]{};

for (int x = 0; x < user\_count; x++)

{

newusers[x] = users[x];

}

newusers[user\_count] = new User{ newuser };

user\_count++;

users = newusers;

newusers = nullptr;

}

bool SignUp(const User& newuser)

{

if (GetUserByUsername(newuser.GetUsername()) == nullptr)

{

AddUserToDatabase(newuser);

return true;

}

return false;

}

bool SignIn(const string& username, const string& password)

{

cin.ignore();

cin.clear();

system("cls");

User\* user = GetUserByUsername(username);

if (user == nullptr)

{

return false;

}

else

{

for (int x = 0; x < user\_count; x++)

{

if (user->GetPassword() == password)

{

return true;

}

}

}

}

int GetUserCount() const

{

return user\_count;

}

User\* GetUserByUsername(const string& username)

{

for (int x = 0; x < user\_count; x++)

{

if (users[x]->GetUsername() == username)

{

return users[x];

}

}

return nullptr;

}

void ShowAllUsers() const

{

system("cls");

cout << " =========== A L L U S E R S =========== " << endl;

for (int x = 0; x < user\_count; x++)

{

users[x]->Show();

}

}

~Database()

{

for (int x = 0; x < user\_count; x++)

{

delete[]users[x];

}

delete[]users;

}

};

void ShowMenu()

{

cout << " ======== M A I N M E N U ======== " << endl;

cout << " Sign In 1 " << endl;

cout << " Sign Up 2 " << endl;

cout << " Show All Users 3 " << endl;

cout << " ================================== " << endl;

}

int main()

{

Database d;

User u1("John Johnlu", "John123", "john\_222");

User u2("Tofiq Tofiqli", "Tofiq333", "tofiq\_333");

User u3("Rafiq Rafiqli", "Rafiq", "rafiq111");

d.AddUserToDatabase(u1);

d.AddUserToDatabase(u2);

d.AddUserToDatabase(u3);

while (true)

{

system("cls");

ShowMenu();

cout << "\n Enter your choice : ";

int choice = 0;

cin >> choice;

if (choice == 1)

{

cin.ignore();

cin.clear();

system("cls");

string username;

cout << " Enter your username : ";

getline(cin, username);

string password;

cout << " Enter your password : ";

d.SetPassword(password);

if (!d.SignIn(username, password))

{

cout << " User does not exist!" << endl;

}

else

{

system("cls");

cout << "\n Welcome" << endl;

system("pause");

}

}

else if (choice == 2)

{

User newuser = d.GetNewUser();

if (!d.SignUp(newuser))

{

cout << "This name has alreay taken!" << endl;

Sleep(3000);

}

else

{

cout << " Your account was successfully created!" << endl;

}

}

else if (choice == 3)

{

d.ShowAllUsers();

system("pause");

}

else

{

system("cls");

cout << "Incorrect Input!" << endl;

Sleep(3000);

}

}

return 0;

}