ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА

ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



ЗВІТ

про виконання лабораторної роботи № 9,3

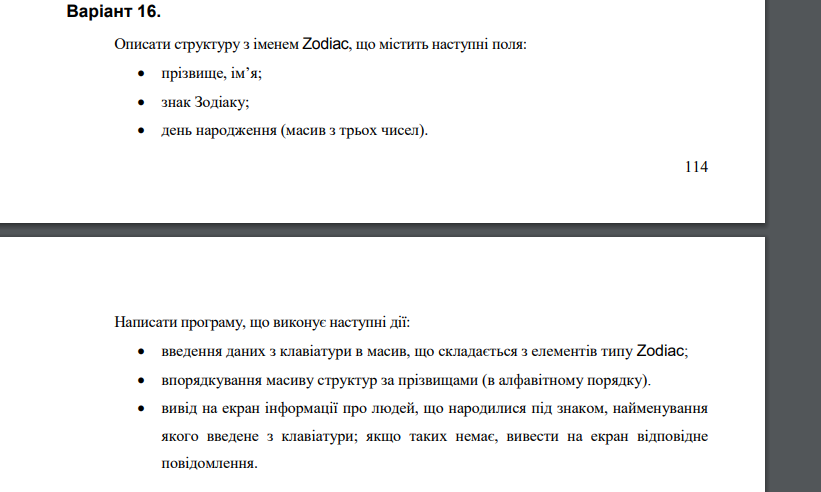
«Опрацювання масиву структур»

з дисципліни «Алгоритмізація та програмування»

студентки групи ІТ-12

Прендкович Ірини Віталіївни

Умова завдання



Відповідь

#include <iostream>

#include <iomanip>

#include <string>

#define TESTING

using namespace std;

#pragma pack(push, 1)

struct Zodiac

{

string surname;

string name;

string zodiac\_sign;

unsigned short b\_day[3];

};

#pragma pack(pop)

void Create(Zodiac\* S, const int N);

void Print(Zodiac\* S, const int N);

void sort(Zodiac\* S, const int N);

void search(Zodiac\* S, const int N, const string f\_surname);

#ifdef TESTING

int main()

{

int N;

cout << "Enter the number of people: "; cin >> N;

Zodiac\* S = new Zodiac[N];

string f\_surname;

int MenuItem;

while (true)

{

cout << endl << endl << endl;

cout << "Select an action: " << endl << endl;

cout << "[1] - keyboard input" << endl;

cout << "[2] - display of data" << endl;

cout << "[3] - physical ordering of data" << endl;

cout << "[4] - binary search for the product by name" << endl;

cout << "0 - exit and exit of the program" << endl;

cout << "Enter the value: ";

cin >> MenuItem;

cout << endl;

switch (MenuItem)

{

case 1:

Create(S, N);

break;

case 2:

Print(S, N);

break;

case 3:

sort(S, N);

Print(S, N);

break;

case 4:

cout << "Enter the search keys: " << endl;

cout << "Zodiak" << endl;

cin.get();

cin.sync();

getline(cin, f\_surname); cout << endl;

search(S, N, f\_surname);

break;

}

if (MenuItem == 0) break;

}

return 0;

}

#endif

void Create(Zodiac\* S, const int N)

{

for (int i = 0; i < N; i++)

{

int speciality;

cout << "People #: " << i + 1 << endl;

cin.get();

cin.sync();

cout << "Surname: "; getline(cin, S[i].surname); cout << endl;

cout << "Name: "; getline(cin, S[i].name); cout << endl;

cout << "Zodiac: "; getline(cin, S[i].zodiac\_sign); cout << endl;

do

{

cout << "Day of birthday: "; cin >> S[i].b\_day[0]; cout << endl;

} while (S[i].b\_day[0] < 1 || S[i].b\_day[0] > 31);

do

{

cout << "Month: "; cin >> S[i].b\_day[1]; cout << endl;

} while (S[i].b\_day[1] < 1 || S[i].b\_day[1] > 12);

cout << "Year: "; cin >> S[i].b\_day[2]; cout << endl;

cout << endl;

}

}

void Print(Zodiac\* S, const int N)

{

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

cout << "| # | Surname | Name | Zodiac | Day | Month | Year |" << endl;

cout << "-----------------------------------------------------------------------------------------" << endl;

for (int i = 0; i < N; i++)

{

cout << "|" << setw(3) << i + 1 << setw(3);

cout << "|" << setw(4) << S[i].surname << setw(8);

cout << "|" << setw(6) << S[i].name << setw(3);

cout << "|" << setw(6) << S[i].zodiac\_sign << setw(6);

cout << "|" << setw(6) << S[i].b\_day[0] << setw(12);

cout << "|" << setw(6) << S[i].b\_day[1] << setw(6);

cout << "|" << setw(10) << S[i].b\_day[2] << setw(6) << "|" << endl;

}

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

}

void sort(Zodiac\* S, int N)

{

Zodiac tmp;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N - i - 1; j++)

{

if (S[j].surname > S[j + 1].surname)

{

tmp = S[j];

S[j] = S[j + 1];

S[j + 1] = tmp;

}

}

}

}

void search(Zodiac\* S, const int N, const string f\_zodiac\_sign)

{

int flag = 0;

for (int i = 0; i < N; i++)

{

if (S[i].zodiac\_sign == f\_zodiac\_sign)

{

flag = 1;

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

cout << "| # | Surname | Name | Zodiac | Day | Month | Year |" << endl;

cout << "-----------------------------------------------------------------------------------------" << endl;

cout << "|" << setw(3) << i + 1 << setw(3);

cout << "|" << setw(4) << S[i].surname << setw(8);

cout << "|" << setw(6) << S[i].name << setw(3);

cout << "|" << setw(6) << S[i].zodiac\_sign << setw(6);

cout << "|" << setw(6) << S[i].b\_day[0] << setw(12);

cout << "|" << setw(6) << S[i].b\_day[1] << setw(6);

cout << "|" << setw(10) << S[i].b\_day[2] << setw(6) << "|" << endl;

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

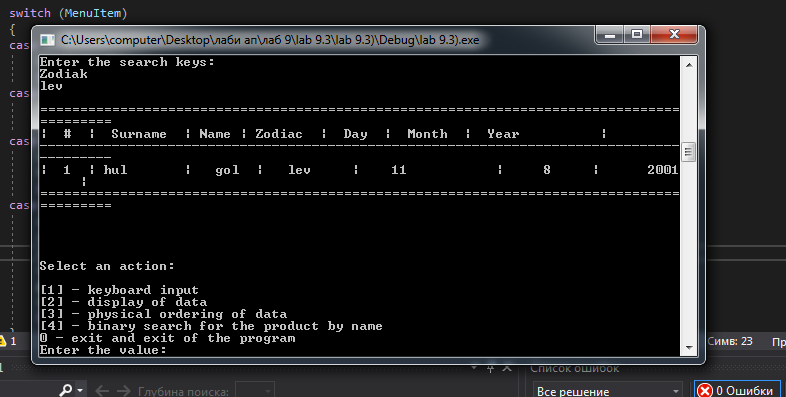
}

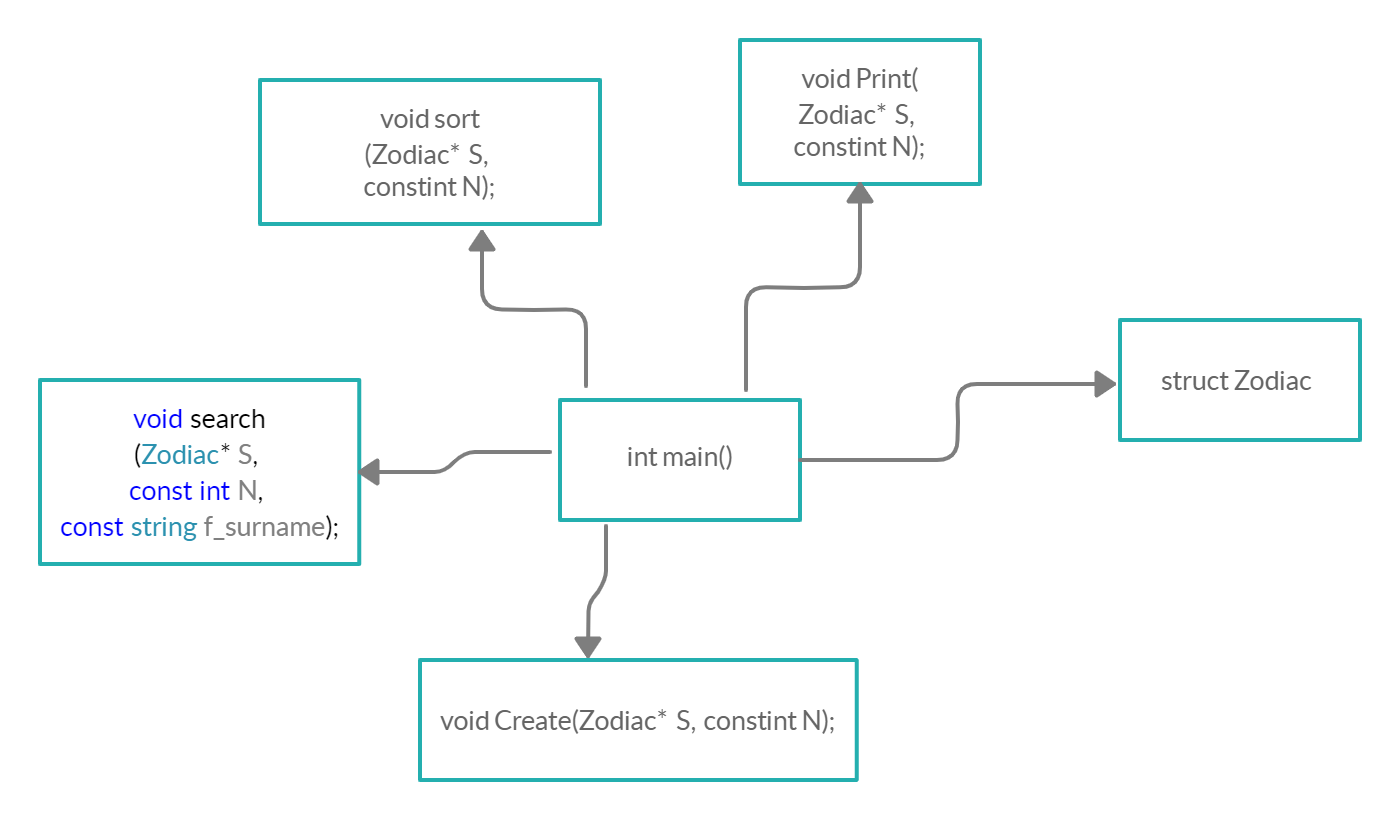
}

if (!flag)

cout << "not found" << endl;

}





<https://github.com/irynaprendkovychitis2020/lab-9.3>

Висновок

На цій лабораторній я навчилася опрацьовувати масив структур з об’єднаннями.