**Лабораторна робота №4**

**Розробка та реалізація програм для роботи зі структурами**

**Ціль роботи**: оволодіння навичками складання програм для роботи зі сруктурами.

**Завдання**

**Завдання 4.1.** Нехай є наступний фрагмент програми. Поясніть, яку задачу реалізує вказаний фрагмент

*Розв’язання*

#include <iostream>

#include <cstring>

#include <cctype>

using namespace std;

const int N = 4;

struct myType {

char fff[20]; char aaa[20]; int b; c;

};

void printStruct(myType st[], int n) {

for (int i = 0; i < n; i++) {

cout << " fff: " << st[i].fff << " aaa: " << st[i].aaa;

cout << " b : " << st[i].b << " c : " << st[i].c << endl;

}

}

int main() {

myType st[] = {

{ "assdss", "ssdhshf",4, 4.5 },

{ "jherhhgb", "xczvfdg",9, 2.4 },

{ "xcvfd", "hhgeryt",5, 4.1 },

{ "jrhjdfd", "tyyyyy",7, 3.5 }

};

printStruct(st, N);

double k = 0;

char c = 'd';

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

if (strchr(st[i].fff, c)) k++;

//Увеличивание k, если в fff какой-то структуры находится 'd'

//(в первых N структурах)

}

. . .

Фрагмент программы считает, в скольки структурах (первых N) в поле fff есть символ 'd'. После выполения k = 3.

**Завдання 4.2.** Створити структуру. Визначити функції, що створюють масив структур. Завдати критерій відбору.

*Розв’язання*

1. Постановка задачі:

Структура "Держава":

− назва;

− столиця;

− чисельність населення;

− займана площа.

Створити масив структур. Вивести:

a) список держав, які мають задану чисельність населення;

b) список держав, що мають площу більшу за вказану і чисельність

населення у вказаному діапазоні;

c) список держав, в алфавітному порядку.

1. Алгоритм розв’язання задачі:

Алгоритм розв’язання задачі можна представити у вигляді такої послідовності дій:

1. Створити массив для країн.
2. Заповнинення массиву данними клаїн (з файлу input.txt; формат даних у файлі: у першому рядку – кількість країн, у наступних данні країн у форматі «назва столиця населення площа»)
3. Сортування країн
4. Вивід країн (відсортованих)
5. Вивід країн з населенням від min до max (де min, max – вводить користувач)
6. Вивід країн з населенням більше вказанного користувачем значення і площею від min до max.
7. **Текст програми**

#include <iostream>

#include <fstream>

#include <string>

#include <windows.h>

using namespace std;

struct country {

string name;

string capital;

long long population;

int S;

void print() {

cout << name << "("

<< "столица: " << capital

<< "; население: " << population

<< "; площадь: " << S << " кв. км)" << endl;

}

};

void writeCountry(country \*countries, int len, ifstream &fin);

void printCountry(country \*countries, int len);

void quickSort(country \*arr, int l, int r);

void printLine(char symbol = '-', short count = 50);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

ifstream fin("input.txt");

int len = 0; //Количество стран, хранящихся в файле

fin >> len;

country \*countries = new country[len];

writeCountry(countries, len, fin);

fin.close();

long long min, max, tmp;

// - - -

cout << endl << "Все страны в алфавитном порядке: " << endl;

printLine();

quickSort(countries, 0, len - 1);

printCountry(countries, len);

printLine();

// - - -

cout << endl << "Страны, с наcелением от: "; cin >> min;

cout << "До: "; cin >> max;

printLine();

for (int i = 0; i < len; i++) {

country item = countries[i];

if (item.population >= min && item.population <= max) {

countries[i].print();

}

}

printLine();

// - - -

cout << endl << "Страны, с населением больше: "; cin >> tmp;

cout << "И площей от: "; cin >> min;

cout << "До: "; cin >> max;

printLine();

for (int i = 0; i < len; i++) {

country item = countries[i];

if (item.population > tmp && item.S >= min && item.S <= max) {

countries[i].print();

}

}

printLine();

system("pause");

return 0;

}

void writeCountry(country \*countries, int len, ifstream &fin) {

for (int i = 0; i < len; i++) {

countries[i] = {};

fin >> countries[i].name;

fin >> countries[i].capital;

fin >> countries[i].population;

fin >> countries[i].S;

}

}

void printCountry(country \*countries, int len) {

for (int i = 0; i < len; i++) {

countries[i].print();

}

}

void printLine(char symbol, short count) {

for (int i = 0; i < count; i++) {

cout << symbol;

}

cout << endl;

}

void quickSort(country \*arr, int l, int r) {

int i = l, j = r;

string middle = arr[(i + j) / 2].name;

do {

while (middle > arr[i].name) i++;

while (middle < arr[j].name) j--;

if (i <= j) {

swap(arr[i], arr[j]);

i++; j--;

}

} while (i < j);

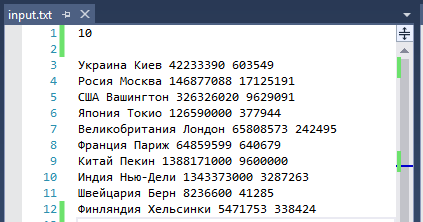
if (i < r) quickSort(arr, i, r);

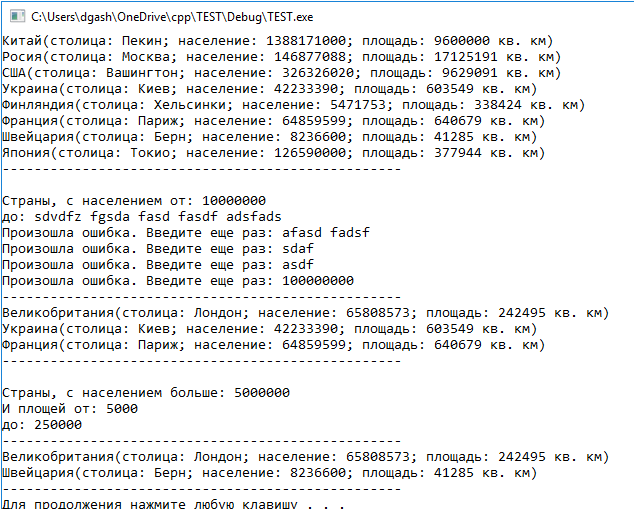
if (j > l) quickSort(arr, l, j);

}

1. Результат виконання програми:

Input.txt:



****

**Висновок:** на цій лабораторній роботі я оволодів навичками складання програм для роботи зі структурами.