МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Липецкий Государственный Технический Университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа

по программированию №5

“Обработка элементов структур”

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Станиславчук С. М.

(подпись, дата)

Группа АС-21-1

Руководитель

Доцент, кандидат наук \_\_\_\_\_\_\_\_\_\_\_\_ Харитоненко А. А.

(подпись, дата)

Липецк 2021 г.

**Содержание:**

1. Титульный лист

2. Цель работы

3. Задание

4. Блок-схема алгоритма

5. Текст программы

6. Пример выполнения

2. Цель работы.

Освоить работу с массивами, элементами которых являются переменные

структурного типа данных языка С.

3. Задание

Программа должна уметь:

1. Создание и ввод элементов массива структур ( Дом { Номер дома (unsigned), улица (массив char), количество этажей (unsigned) } )

2. Установка/получение значений полей элемента массива структур по

заданному индексу в массиве (индекс вводится пользователем).

3. Поиск элемента массива структур, в котором значение выбранного поля

совпадает со значением поля, введенным пользователем (номер поля, по

которому осуществляется поиск, вводится пользователем).

4. Обмен значений соответствующих полей двух заданных элементов массива.

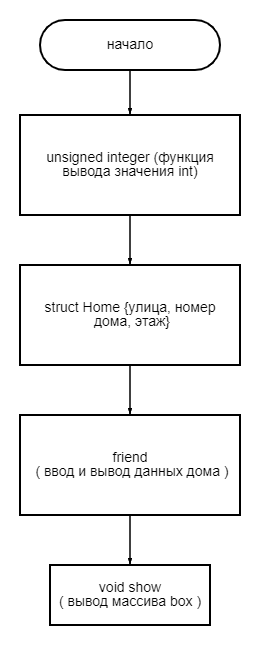
структур (индексы элементов вводятся пользователем).

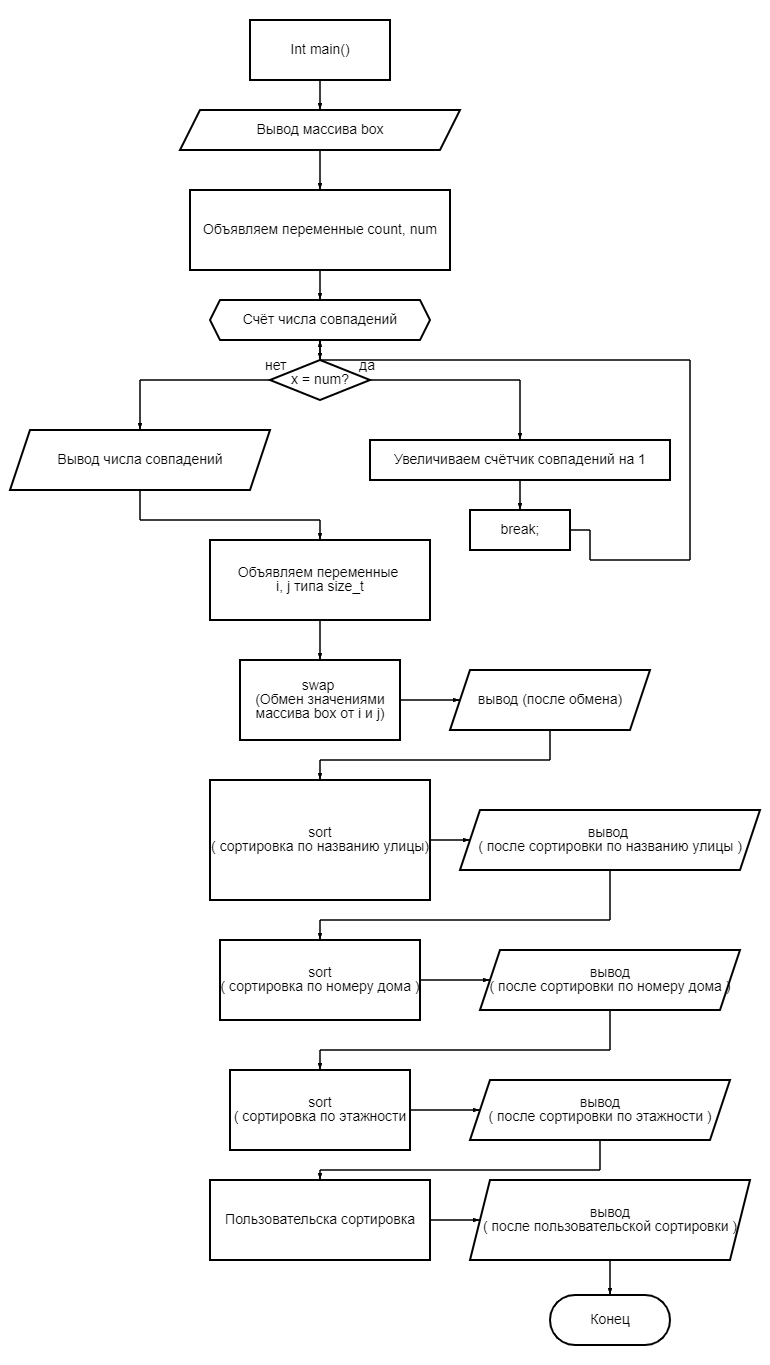
5. Упорядочение элементов массива структур по убыванию

значений заданного поля.

6. Вывод одного элемента массива структур или диапазона элементов (индекс или диапазон индексов вводится пользователем).

4. Блок-схема





5. Текст программы

#include <algorithm>

#include <iostream>

#include <iomanip>

#include <cstring>

#include <string>

using namespace std;

unsigned integer(istream& inp, const char\* msg)

{

cout << msg;

unsigned value;

inp >> value;

inp.ignore(inp.rdbuf()->in\_avail());

return value;

}

struct Home

{

unsigned number;

unsigned level;

char name[0x30];

Home() : number(0), level(0)

{

memset(name, 0, sizeof(name));

}

friend istream& operator>>(istream& inp, Home& h)

{

cout << "Название улицы: ";

inp.getline(h.name, size(h.name));

h.number = integer(inp, "Номер дома: ");

h.level = integer(inp, "Количество этажей: ");

return inp;

}

friend ostream& operator<<(ostream& out, const Home& h)

{

return out << "Улица " << h.name << ", дом " << h.number << ", этажей: " << h.level;

}

};

void show(const char\* title, Home\* box, const size\_t n)

{

cout << '\t' << title << '\n';

for (auto i = 0U; i < n; ++i) cout << box[i] << '\n';

puts("");

}

int main()

{

system("chcp 1251 > nul"); // для кириллицы при вводе и выводе

Home box[5];

for (auto& x : box) cin >> x;

show("Введенные данные: ", box, size(box));

auto count = 0;

auto num = integer(cin, "Введите номер дома: ");

for (const auto& x : box)

{

if (x.number == num)

{

cout << x << '\n';

++count;

break;

}

}

cout << "Итого: найдено " << count << " совпадений\n";

size\_t i, j;

do

{

i = integer(cin, "Введите первый индекс: ");

} while (!i || i >= size(box));

do

{

j = integer(cin, "Введите второй индекс: ");

} while (!j || j >= size(box) || j == i);

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

show("После обмена: ", box, size(box));

sort(begin(box), end(box), [](const Home& a, const Home& b)

{

return string(a.name) > string(b.name);

});

show("После сортировки по названию: ", box, size(box));

sort(begin(box), end(box), [](const Home& a, const Home& b)

{

return a.number > b.number;

});

show("После сортировки по номеру дома: ", box, size(box));

sort(begin(box), end(box), [](const Home& a, const Home& b)

{

return a.level > b.level;

});

show("После сортировки по этажности: ", box, size(box));

do

{

i = integer(cin, "Введите индекс: ");

} while (!i || i >= size(box));

cout << box[i] << '\n';

do

{

i = integer(cin, "Введите первый индекс диапазона: ");

} while (!i || i >= size(box));

do

{

j = integer(cin, "Введите второй индекс диапазона: ");

} while (!j || j >= size(box) || j == i);

for (auto f = i; f <= j; ++f) cout << box[f] << '\n';

system("pause > nul");

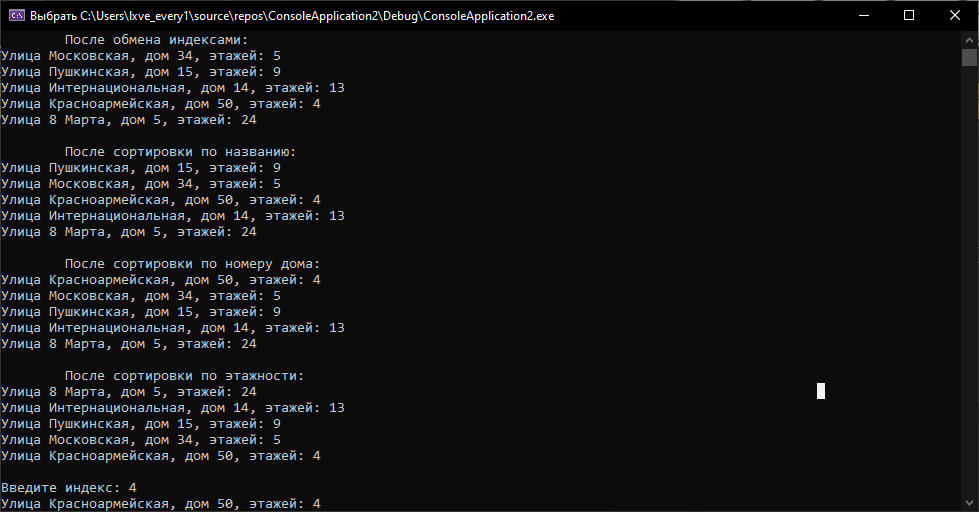
return 0;

}

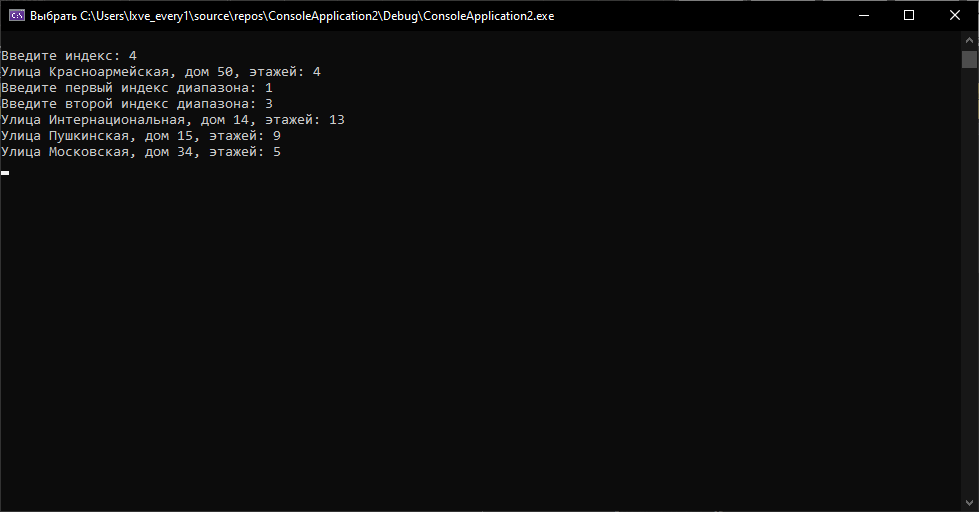
6. Пример выполнения

1) Вводим данные  

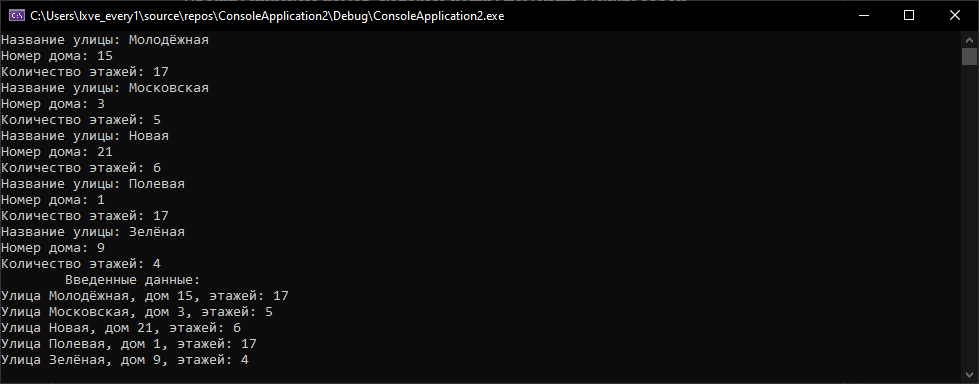

Вводим индексы домов, которые хотим поменять между собой



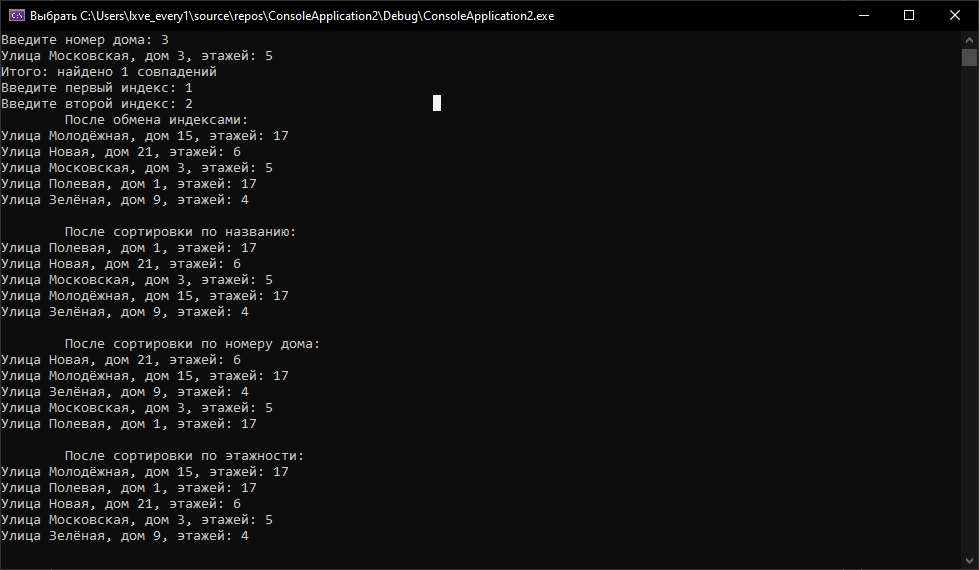
Выбираем диапозон индексов, который хотим вывести на экран



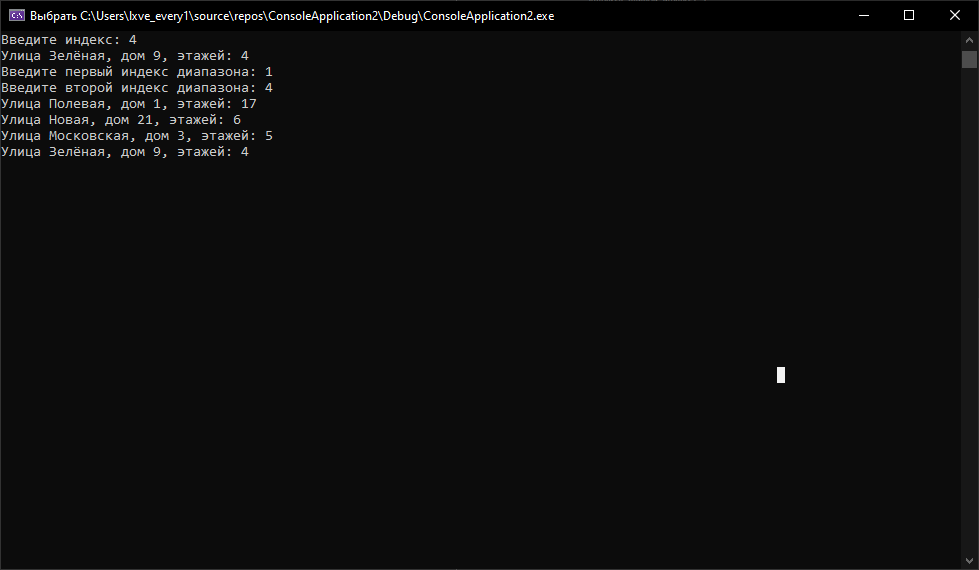
Программа выполнилась успешно.  
2) Вводим данные



Вводим индексы домов, которые хотим поменять между собой



Выбираем диапозон индексов, который хотим вывести на экран



Программа выполнилась успешно.

Вывод: научился работать с массивами, элементами которых являются структурные данные.