Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №15**

Дисциплина: основы алгоритмизации и программирования

Тема: “Сортировки Шелла и Хоара"

**Вар.20**

Выполнил работу

студент группы ИВТ-20-2Б

Кузнецов Н.Д

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2020

**Цель задачи**

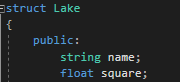
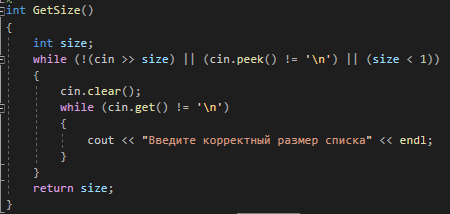
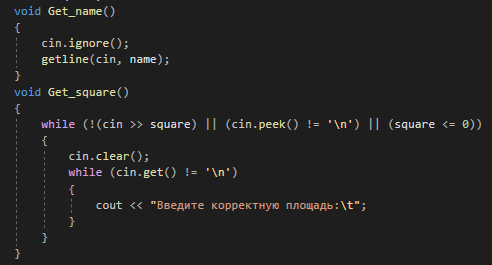
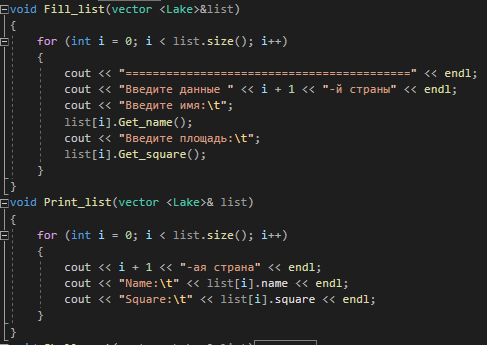
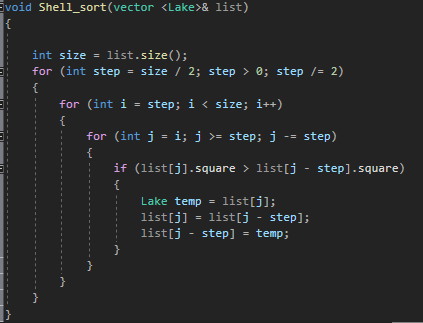
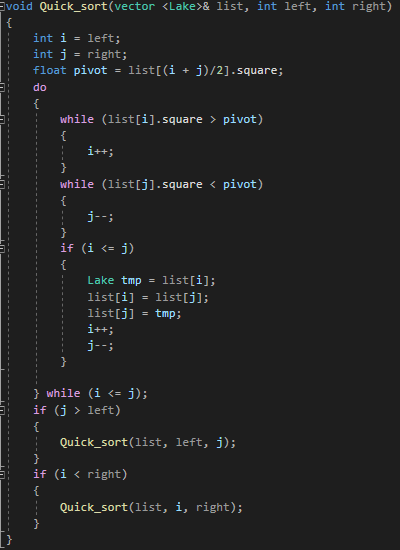
1) Получить практические навыки работы с сортировками;  
Для достижения поставленной цели необходимо решить следующие задачи:

* Разработать функцию сортировки методом Шелла.
* Разработать функцию сортировки методом Хоара.
* Организовать ввод-вывод данных.
* Разработать программу.

**Постановка задачи**

Дан список озер и их площади. Отсортировать список по убыванию площади.

**Анализ задачи**

1. Для решения задачи используются типы данных **float**, **int**, класс **vector** и класс **string**.
   1. Поле имени структуры имеет тип данных string, поле площади - float.  
      
   2. Для корректного ввода пользователем размера списка разработана функция GetSize, которая не принимает параметров и возвращает тип данных **int**.  
      
   3. Для хранения данных объектов структуры используется vector.  
      
   4. Функции заполнения и вывода элементов вектора не возвращают значений и принимают в качестве параметра ссылку на первый элемент вектора.  
      
   5. Функция сортировки Шелла принимает и возвращают значения аналогично п.1.4.  
      
   6. Функция сортировки Хоара принимает ссылку на первый элемент вектора, левую и правую границы массива.  
      
2. Для решения задачи необходимо:
   1. Получить корректный размер списка от пользователя. (см.п.1.2).
   2. Разработать методы структуры, которые будут вводить данные в поля объекта структуры (Get\_name, Get\_square).
   3. Реализовать функции ввода-вывода данных из списка (Fill\_list, Print\_list).  
      
   4. Разработать функцию сортировки Шелла (Shell\_sort).  
      
   5. Разработать функцию сортировки Хоара (Quick\_sort).  
      
3. Для решения задачи данные были представлены в следующем виде:
   1. Для сортировки списка с озерами данные представлены в виде vector’а с типом данных Lake.
   2. Для хранения информации о озере реализована структура Lake.
4. Структура Lake имеет 2 поля: поле имени (string), поле площади (float).
5. Для ввода вывода данных используются следующие операторы:
   1. cin и cout в функциях заполнения, вывода массива, так же, в функции main и методах структуры.
   2. getline используется в методе ввода поля строки (Get\_name).
6. Поставленные задача будут решены следующими действиями:
   1. Сортировка методом Шелла реализована с помощью вложенных циклов for. Первый цикл уменьшает так называемый шаг(step), второй отвечает за разбиение массива на списки, отстающих на step, 3 цикл, вложенный во второй, отвечает за саму сортировку.
   2. Сортировка методом Хоара реализована с помощью цикла do while, в который вложено еще два цикла while, которые отвечают за сдвиг индексов, указывающих на текущие элементы. В случае выхода из вложенных индексов, i и j элемент массива меняются. Цикл do while работает до тех пор, пока i-ый индекс <= j-ому индексу. В случае, если j-ый индекс больше левой границы, то функция сортировки вызывается рекурсивно, с новым индексом правой границы (j). Если же i < right, то функция вызывается рекурсивно с новым значением левой границы (i). Для деления массива на двое каждый раз используется переменная pivot, в которой хранится значение элемента с индексом (left+right)/2.

**Код программы**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

struct Lake

{

public:

string name;

float square;

void Get\_name()

{

cin.ignore();

getline(cin, name);

}

void Get\_square()

{

while (!(cin >> square) || (cin.peek() != '\n') || (square <= 0))

{

cin.clear();

while (cin.get() != '\n')

{

cout << "Введите корректную площадь:\t";

}

}

}

};

int GetSize()

{

int size;

while (!(cin >> size) || (cin.peek() != '\n') || (size < 1))

{

cin.clear();

while (cin.get() != '\n')

{

cout << "Введите корректный размер списка" << endl;

}

}

return size;

}

void Fill\_list(vector <Lake>&list)

{

for (int i = 0; i < list.size(); i++)

{

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

cout << "Введите данные " << i + 1 << "-й страны" << endl;

cout << "Введите имя:\t";

list[i].Get\_name();

cout << "Введите площадь:\t";

list[i].Get\_square();

}

}

void Print\_list(vector <Lake>& list)

{

for (int i = 0; i < list.size(); i++)

{

cout << i + 1 << "-ая страна" << endl;

cout << "Name:\t" << list[i].name << endl;

cout << "Square:\t" << list[i].square << endl;

}

}

void Shell\_sort(vector <Lake>& list)

{

int size = list.size();

for (int step = size / 2; step > 0; step /= 2)

{

for (int i = step; i < size; i++)

{

for (int j = i; j >= step; j -= step)

{

if (list[j].square > list[j - step].square)

{

Lake temp = list[j];

list[j] = list[j - step];

list[j - step] = temp;

}

}

}

}

}

void Quick\_sort(vector <Lake>& list, int left, int right)

{

int i = left;

int j = right;

float pivot = list[(i + j)/2].square;

do

{

while (list[i].square > pivot)

{

i++;

}

while (list[j].square < pivot)

{

j--;

}

if (i <= j)

{

Lake tmp = list[i];

list[i] = list[j];

list[j] = tmp;

i++;

j--;

}

} while (i <= j);

if (j > left)

{

Quick\_sort(list, left, j);

}

if (i < right)

{

Quick\_sort(list, i, right);

}

}

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Введите размер списка стран:\t";

int size = GetSize();

vector <Lake> list (size);

Fill\_list(list);

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

Print\_list(list);

cout << "==========================================" << endl << "Сортировка шелла:" << endl;

Shell\_sort(list);

Print\_list(list);

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

Fill\_list(list);

Print\_list(list);

cout << "==========================================" << endl << "Быстрая сортировка Хоара:" << endl;

//Quick\_sort(list, 0, list.size() - 1);

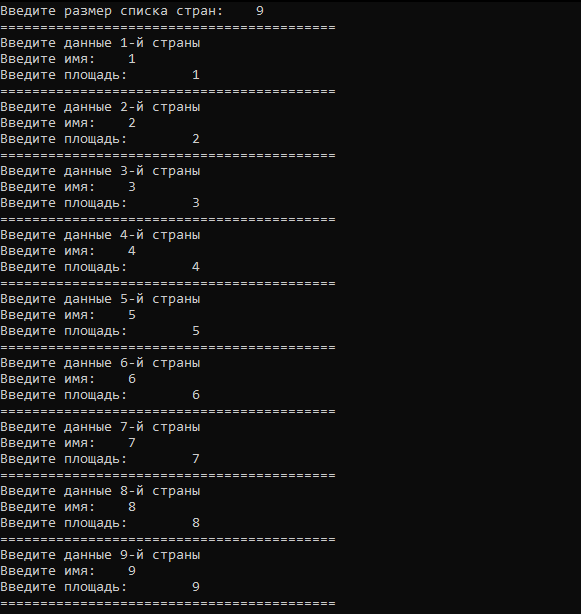
Quick\_sort\_stack(list, 0, list.size() - 1);

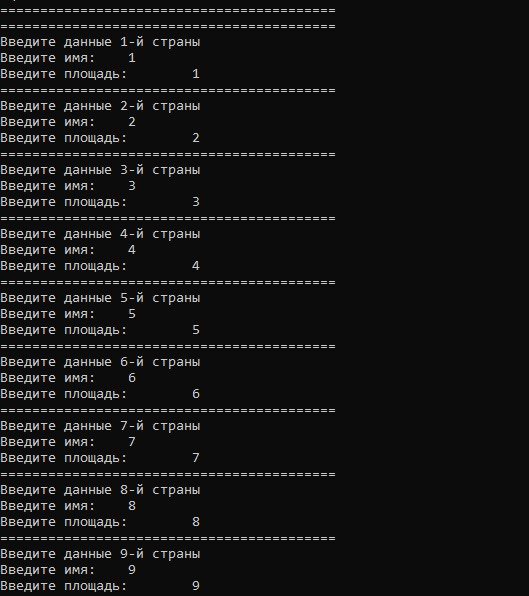
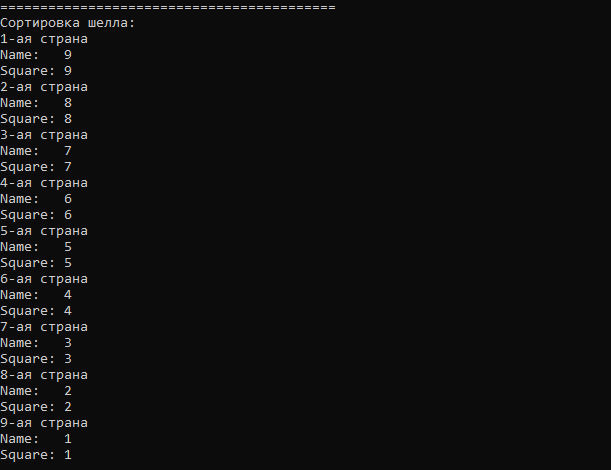
Print\_list(list);

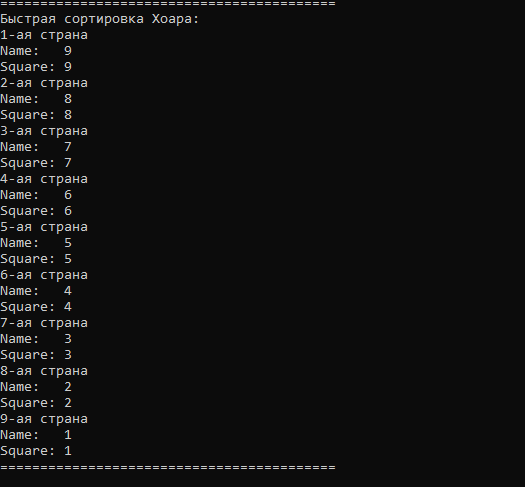
cout << "==========================================" << endl;

}

**Результат выполнения**







**Блок-схема**

