Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №15**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Методы внутренней сортировки массивов быстрая сортировка, сортировка подсчётом

Вариант 19

Выполнил:

Студент группы РИС-20-1б

Ремянников Александр Владимирович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Получить опыт работы с методами сортировки.

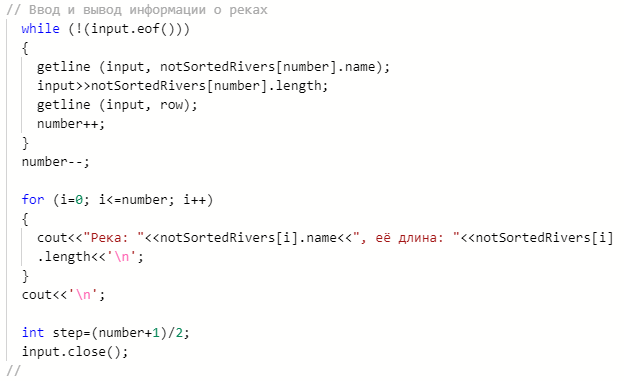
**Постановка задачи**

(Вариант 19) Отсортировать заданный массив структур методом Шелла и методом Хоара.

**Анализ задачи**

**1.** Для решения задачи необходимо:

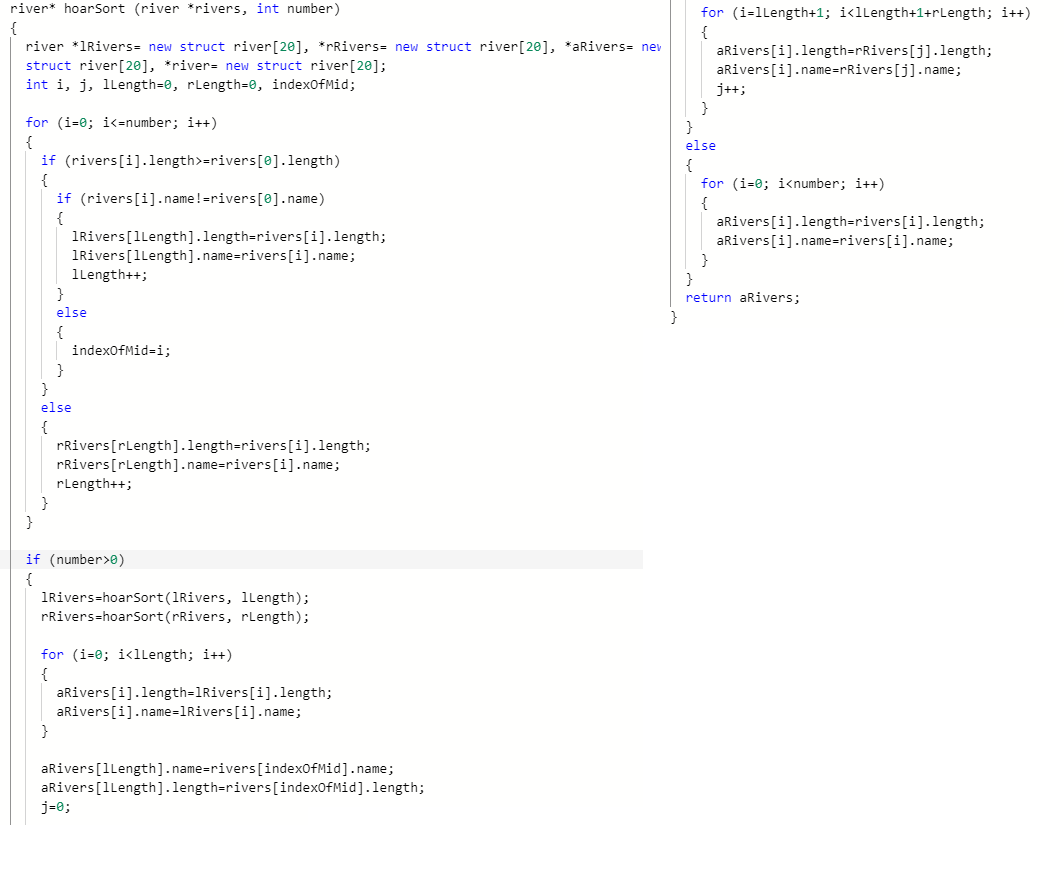
**1.1.** Считать информацию о реках из файла и вывести ее в консоль.



**1.2.** Реализовать код для сортировки массива методом Шелла.

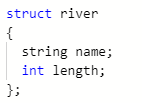


**1.3.** Реализовать код для сортировки массива методом Хоара.



**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Для хранения информации о реках используется массив структур river.



**3.** Для решения задачи данные были представлены в следующем виде:

**3.1.** Для успешной сортировки поле length было выбрано в качестве ключа сортировки.

**4.** Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.** Для операции ввода использовались операторы cin и getline.



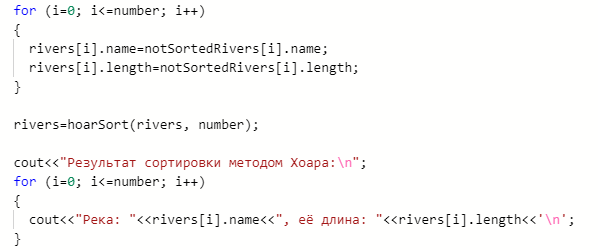
**4.2.** Для операции вывода использовался оператор cout.



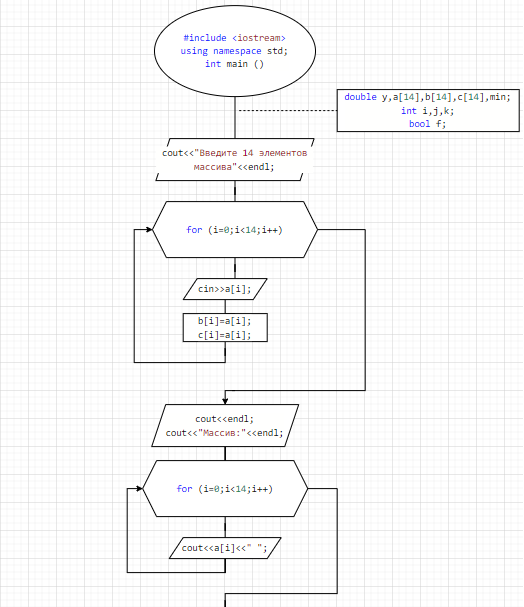
**5.** Поставленные задачи будут решены следующими действиями:

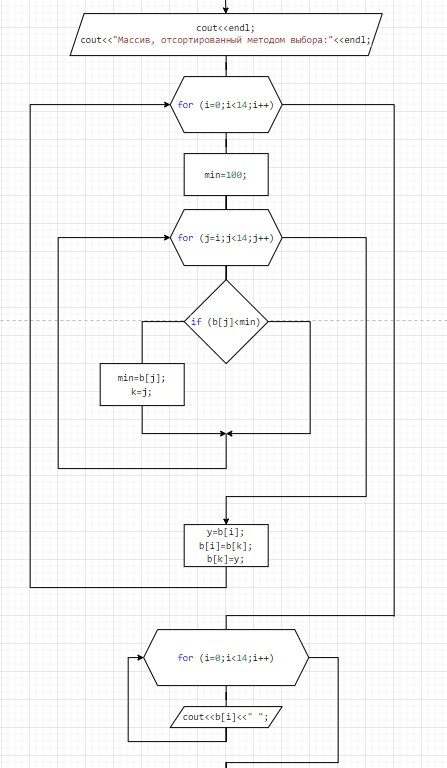
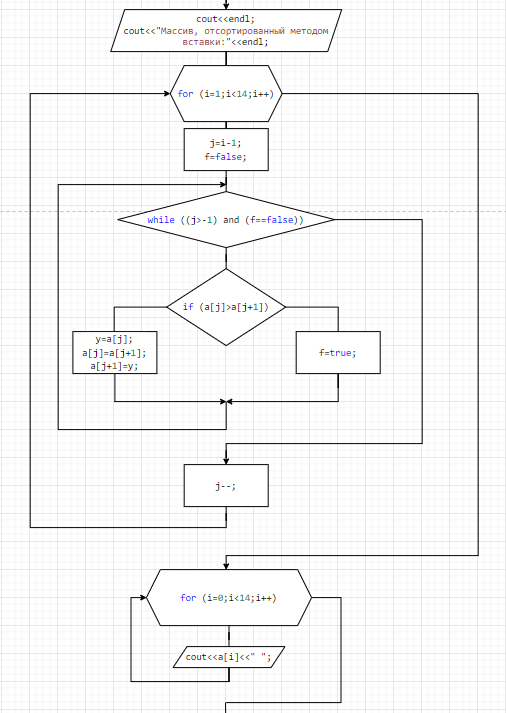
**5.1.** Будет применен код для сортировки методом Шелла.

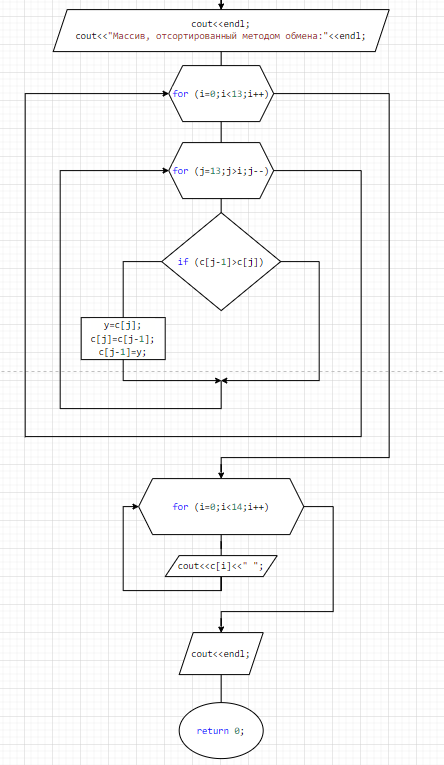
**5.2.** Будет применен код для сортировки методом Хоара.

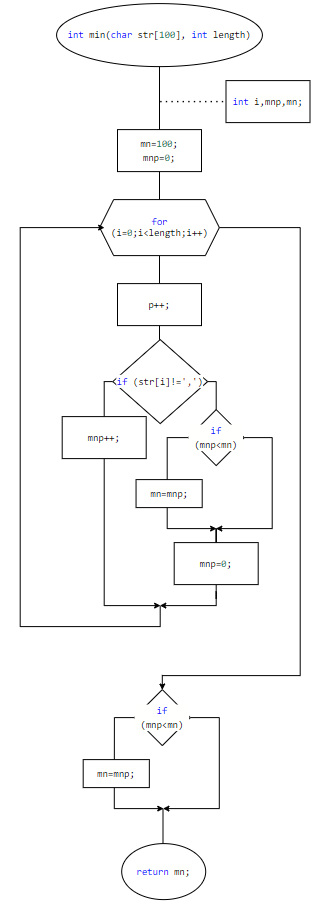


**Блок-схема программы**

****

****

****



**Решение**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct river

{

string name;

int length;

};

river\* hoarSort (river\*, int);

int main()

{

// Описание переменных

setlocale (LC\_ALL,"Russian");

int number=0, i, j, k, length;

river notSortedRivers[20], \*rivers=new struct river[20];

string row, name;

bool f;

ifstream input ("rivers.txt");

//

// Ввод и вывод информации о реках

while (!(input.eof()))

{

getline (input, notSortedRivers[number].name);

input>>notSortedRivers[number].length;

getline (input, row);

number++;

}

number--;

for (i=0; i<=number; i++)

{

cout<<"Река: "<<notSortedRivers[i].name<<", её длина: "<<notSortedRivers[i].length<<'\n';

}

cout<<'\n';

int step=(number+1)/2;

input.close();

//

// Метод Шелла

for (i=0; i<=number; i++)

{

rivers[i].name=notSortedRivers[i].name;

rivers[i].length=notSortedRivers[i].length;

}

while (step>0)

{

for (i=0; i<step; i++)

{

for (j=i; j<=number-step; j+=step)

{

k=j;

f=false;

while ((k>-1) and (f==false))

{

if (rivers[k].length < rivers[k+step].length)

{

length = rivers[k].length;

rivers[k].length = rivers[k+step].length;

rivers[k+step].length = length;

name = rivers[k].name;

rivers[k].name = rivers[k+step].name;

rivers[k+step].name = name;

}

else

{

f=true;

}

k-=step;

}

}

}

step/=2;

}

cout<<"Результат сортировки методом Шелла:\n";

for (i=0; i<=number; i++)

{

cout<<"Река: "<<rivers[i].name<<", её длина: "<<rivers[i].length<<'\n';

}

cout<<'\n';

//

// Метод Хоара

for (i=0; i<=number; i++)

{

rivers[i].name=notSortedRivers[i].name;

rivers[i].length=notSortedRivers[i].length;

}

rivers=hoarSort(rivers, number);

cout<<"Результат сортировки методом Хоара:\n";

for (i=0; i<=number; i++)

{

cout<<"Река: "<<rivers[i].name<<", её длина: "<<rivers[i].length<<'\n';

}

//

}

river\* hoarSort (river \*rivers, int number)

{

river \*lRivers= new struct river[20], \*rRivers= new struct river[20], \*aRivers= new struct river[20], \*river= new struct river[20];

int i, j, lLength=0, rLength=0, indexOfMid;

for (i=0; i<=number; i++)

{

if (rivers[i].length>=rivers[0].length)

{

if (rivers[i].name!=rivers[0].name)

{

lRivers[lLength].length=rivers[i].length;

lRivers[lLength].name=rivers[i].name;

lLength++;

}

else

{

indexOfMid=i;

}

}

else

{

rRivers[rLength].length=rivers[i].length;

rRivers[rLength].name=rivers[i].name;

rLength++;

}

}

if (number>0)

{

lRivers=hoarSort(lRivers, lLength);

rRivers=hoarSort(rRivers, rLength);

for (i=0; i<lLength; i++)

{

aRivers[i].length=lRivers[i].length;

aRivers[i].name=lRivers[i].name;

}

aRivers[lLength].name=rivers[indexOfMid].name;

aRivers[lLength].length=rivers[indexOfMid].length;

j=0;

for (i=lLength+1; i<lLength+1+rLength; i++)

{

aRivers[i].length=rRivers[j].length;

aRivers[i].name=rRivers[j].name;

j++;

}

}

else

{

for (i=0; i<number; i++)

{

aRivers[i].length=rivers[i].length;

aRivers[i].name=rivers[i].name;

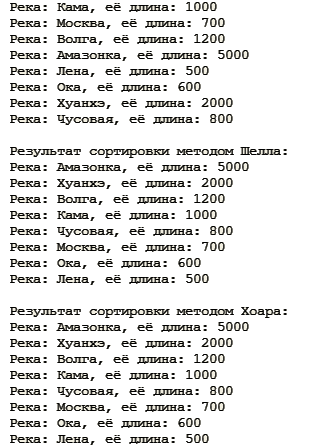
}

}

return aRivers;

}

**Скриншоты результатов работы программы**

****

**GitHub**

<https://github.com/sugarrrfqs/lab15/tree/main>