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.21**

Выполнил работу

студент группы ИВТ-20-2Б

Галинов О.Ю.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2020

**Цель задачи**

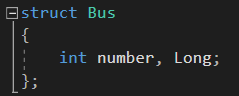
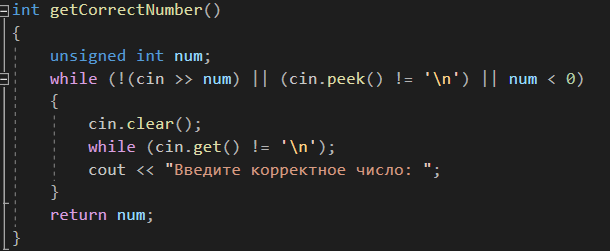
1) Получить практические навыки работы с сортировками;  
Для достижения поставленной цели необходимо решить следующие задачи:

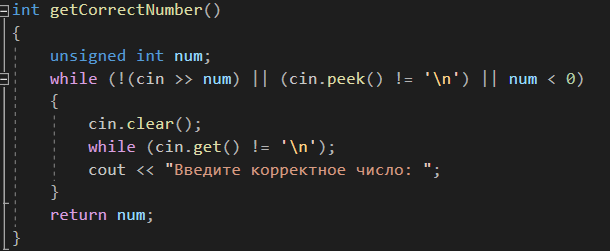
* Разработать функцию сортировки методом Шелла.
* Разработать функцию сортировки методом Хоара.
* Организовать ввод-вывод данных.
* Разработать программу.

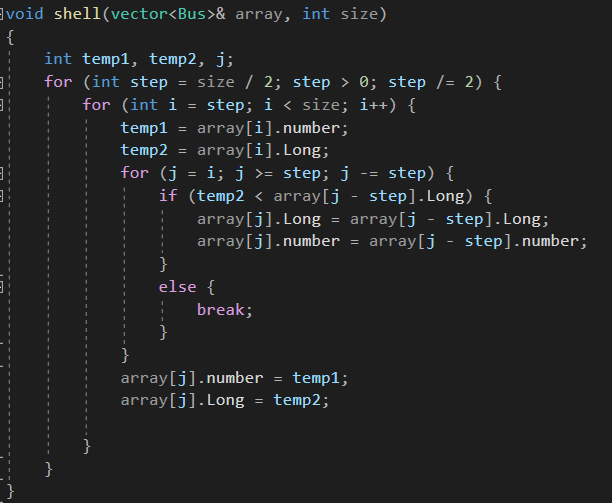
**Постановка задачи**

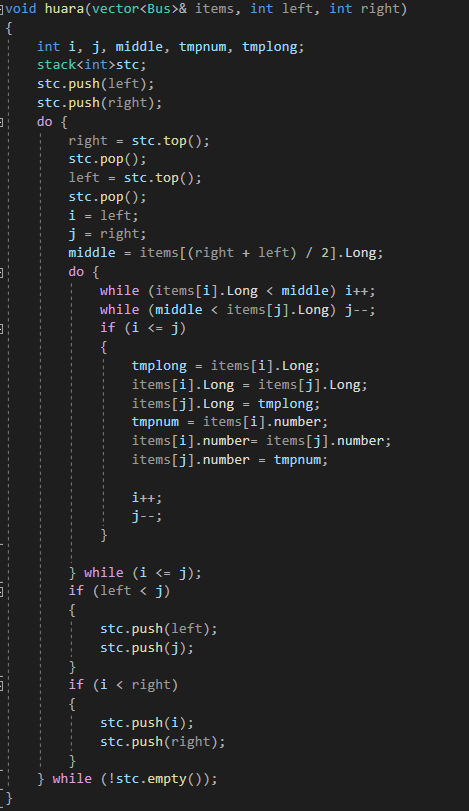
Дан список автобусов и длин их маршрутов, отсортировать длины маршрутов по возрастанию.

**Анализ задачи**

1. Для решения задачи используются тип данных **int** и класс **vector**.
   1. Поле номера структуры имеет тип данных int, поле площади - int.  
      
   2. Для корректного ввода пользователем размера списка разработана функция getCorrectNumber, которая не принимает параметров и возвращает тип данных **int**.  
      
   3. Для хранения данных объектов структуры используется vector.  
      
   4. Функция сортировки Шелла принимает ссылку на первый элемент вектора и его размер -1.  
      
   5. 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).



* 1. Разработать функцию сортировки Шелла (shell).  
     
  2. Разработать функцию сортировки Хоара (huara).



1. 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.
2. Структура Lake имеет 2 поля: поле номера (int), поле длины маршрута (int).
3. Для ввода вывода данных используются следующие операторы:
   1. cin и cout в функции main и методах структуры.
4. 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<stack>

using namespace std;

struct Bus

{

int number, Long;

};

int getCorrectNumber()

{

unsigned int num;

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

{

cin.clear();

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

cout << "Введите корректное число: ";

}

return num;

}

void shell(vector<Bus>& array, int size)

{

int temp1, temp2, j;

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

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

temp1 = array[i].number;

temp2 = array[i].Long;

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

if (temp2 < array[j - step].Long) {

array[j].Long = array[j - step].Long;

array[j].number = array[j - step].number;

}

else {

break;

}

}

array[j].number = temp1;

array[j].Long = temp2;

}

}

}

void huara(vector<Bus>& items, int left, int right)

{

int i, j, middle, tmpnum, tmplong;

stack<int>stc;

stc.push(left);

stc.push(right);

do {

right = stc.top();

stc.pop();

left = stc.top();

stc.pop();

i = left;

j = right;

middle = items[(right + left) / 2].Long;

do {

while (items[i].Long < middle) i++;

while (middle < items[j].Long) j--;

if (i <= j)

{

tmplong = items[i].Long;

items[i].Long = items[j].Long;

items[j].Long = tmplong;

tmpnum = items[i].number;

items[i].number= items[j].number;

items[j].number = tmpnum;

i++;

j--;

}

} while (i <= j);

if (left < j)

{

stc.push(left);

stc.push(j);

}

if (i < right)

{

stc.push(i);

stc.push(right);

}

} while (!stc.empty());

}

int main()

{

int size;

setlocale(LC\_ALL, "rus");

cout << "Введите кол-во элементов: ";

size = getCorrectNumber();

vector <Bus> buses(size);

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

{

cout << "Номер автобуса: ";

cin >> buses[i].number;

cout << "Длина маршрута: ";

cin >> buses[i].Long;

cout << endl;

}

shell(buses, size);

cout << endl;

cout << "Shell: " << endl << endl;

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

{

cout << "Номер автобуса: " << buses[i].number << " - " << "Длина маршрута: " << buses[i].Long << endl;

}

cout << endl << endl;

huara(buses, 0, size-1);

cout << "Hoara:" << endl << endl;

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

{

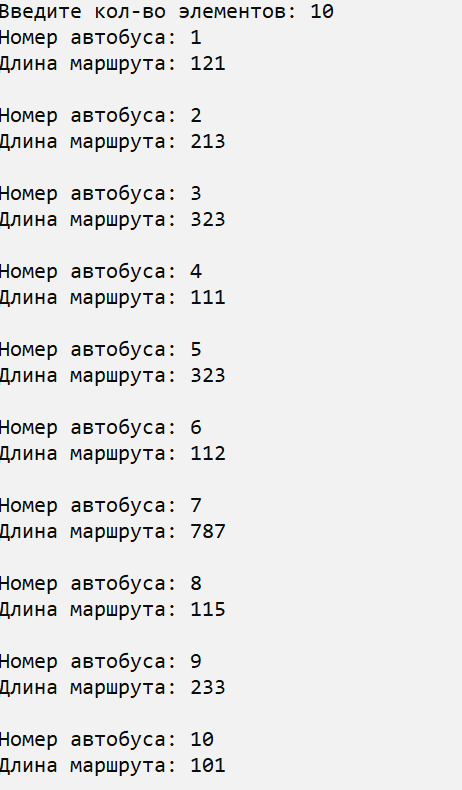
cout << "Номер автобуса: " << buses[i].number << " - " << "Длина маршрута: " << buses[i].Long << endl;

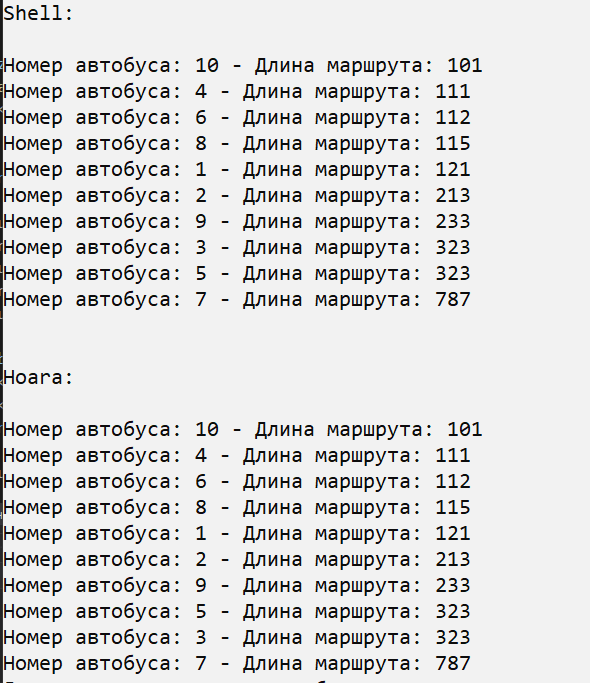
}

system("pause");

}

**Результат выполнения**





**Блок-схема**

