Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы ИВТ-21-1б  
Мифтахов И.Г

Проверил  
ст. п. кафедры ИТАС  
Яруллин Д.В.

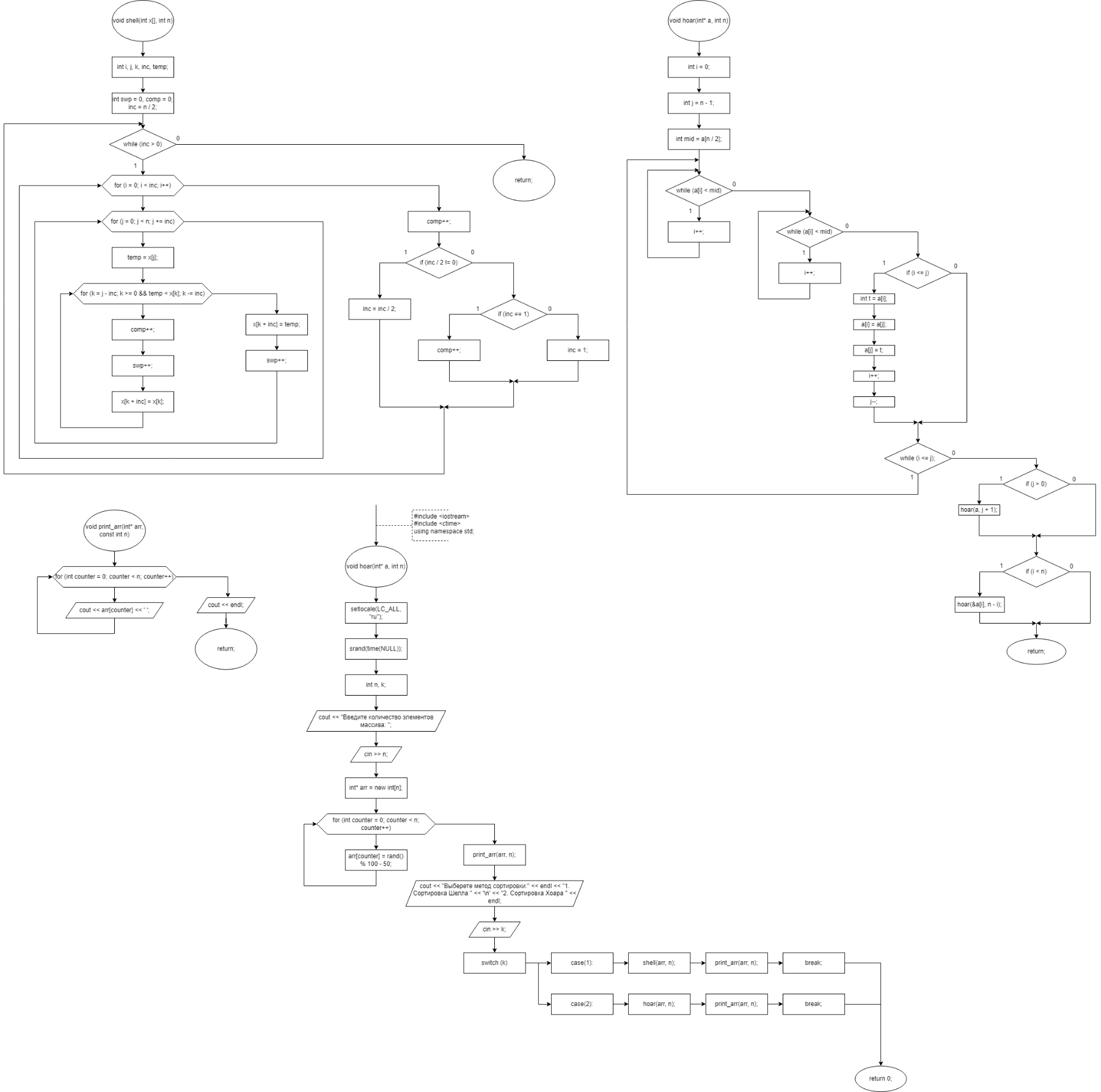
Пермь 2022

Постановка задачи:

Написать сортировки массива методами Шелла и Хоара.

Анализ задачи:

1. Сортировка Шелла
   1. Сортировка Шелла является усовершенствованным вариантом [сортировки вставками](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%B2%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B0%D0%BC%D0%B8). Идея метода Шелла состоит в сравнении элементов, стоящих не только рядом, но и на определённом расстоянии друг от друга.
   2. При сортировке Шелла сначала сравниваются и сортируются между собой значения, стоящие один от другого на некотором расстоянии inc. После этого процедура повторяется для некоторых меньших значений inc, а завершается сортировка Шелла упорядочиванием элементов при inc=1.
2. Сортировка Хоара
   1. Сортировка Хоара является существенно улучшенным вариантом алгоритма «[Пузырьковой сортировки](https://ru.wikipedia.org/wiki/%D0%9F%D1%83%D0%B7%D1%8B%D1%80%D1%8C%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0)» известного в том числе своей низкой эффективностью. Принципиальное отличие состоит в том, что в первую очередь производятся перестановки на наибольшем возможном расстоянии и после каждого прохода элементы делятся на две независимые группы.
   2. Сначала необходимо выбрать из массива элемент, называемый опорным. Это может быть любой из элементов массива.
   3. Сравнить все остальные элементы с опорным и переставить их в массиве так, чтобы разбить массив на три непрерывных отрезка, следующих друг за другом: «элементы меньшие опорного», «равные» и «большие».
   4. Для отрезков «меньших» и «больших» значений выполнить рекурсивно ту же последовательность операций, если длина отрезка больше единицы.
3. Для удобства можно написать функицю, которая выводит массив на экран.

Блок-схема:

Код программы:

#include <iostream>

#include <ctime>

using namespace std;

void print\_arr(int\* arr, const int n)

{

for (int counter = 0; counter < n; counter++)

{

cout << arr[counter] << ' ';

}

cout << endl;

}

void shell(int x[], int n)

{

int i, j, k, inc, temp;

int swp = 0, comp = 0; inc = n / 2;

while (inc > 0)

{

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

{

for (j = 0; j < n; j += inc)

{

temp = x[j];

for (k = j - inc; k >= 0 && temp < x[k]; k -= inc)

{

comp++;

swp++;

x[k + inc] = x[k];

}

x[k + inc] = temp;

swp++;

}

}

comp++;

if (inc / 2 != 0)

inc = inc / 2;

else if (inc == 1)

inc = 0;

else

inc = 1;

}

}

void hoar(int\* a, int n)

{

int i = 0;

int j = n - 1;

int mid = a[n / 2];

do

{

while (a[i] < mid)

{

i++;

}

while (a[j] > mid)

{

j--;

}

if (i <= j)

{

int t = a[i];

a[i] = a[j];

a[j] = t;

i++;

j--;

}

} while (i <= j);

if (j > 0)

{

hoar(a, j + 1);

}

if (i < n)

{

hoar(&a[i], n - i);

}

}

int main()

{

setlocale(LC\_ALL, "ru");

srand(time(NULL));

int n, k;

cout << "Введите количество элементов массива: ";

cin >> n;

int\* arr = new int[n];

for (int counter = 0; counter < n; counter++)

{

arr[counter] = rand() % 100 - 50;

}

print\_arr(arr, n);

cout << "Выберете метод сортировки:" << endl << "1. Сортировка Шелла " << '\n' << "2. Сортировка Хоара " << endl;

cin >> k;

switch (k)

{

case(1):

shell(arr, n);

print\_arr(arr, n);

break;

case(2):

hoar(arr, n);

print\_arr(arr, n);

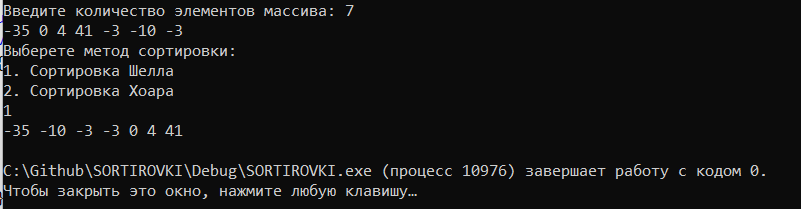
break;

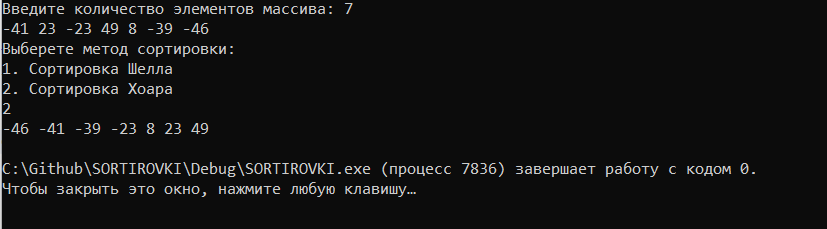
}

return 0;

}

Скриншоты результатов:





Анализ результатов:

Программа работает корректно. Выводится отсортированный массив.