Міністерство освіти і науки України

НТУУ „КПІ”

ТЕФ

**Кафедра АПЕПС**

***Звіт***

***з циклу лабораторних робіт***

***з дисципліни***

***„Основи програмування 2”***

Варіант 22

Виконав: Трухан Д.В.

Студент групи: ***ТВ*-**61

Перевірив викладач: Гусєва І.І.

**Київ – 2017**

Лабораторна робота 4

#include "stdafx.h"

#include <iostream>

#include <iomanip>

using namespace std;

void menu();

int \*\*init(int &n);

void output(int \*\*array, int n);

void SortShell(int \*\*array, int n);

void SortHoora(int \*\*array, int n, int first, int last);

void QuickSort(int \*\*array, int n, int first, int last);

int main()

{

setlocale(0, "");

srand(time\_t(NULL));

menu();

return 0;

}

void menu()

{

int m, n, \*\*array = 0, input;

array = init( n);

output(array, n);

cout << "Меню" << endl;

cout << "1. Выполнить сортировку столбцов с помощью сортировки Шелла " << endl;

cout << "2. Выполнить сортировку столбцов с помощью сортировки Хоора " << endl;

cout << "3. Выполнить сортировку столбцов с помощью сортировки QuickSort " << endl;

cout << "4. Выход из программы"<<endl<<endl;

do

{

cin >> input;

switch (input)

{

case 1:

SortShell(array, n);

cout << "Отсортированный массив имеет вид: " << endl;

output(array, n);

cout << endl << endl;

break;

case 2:

SortHoora(array, n, 0, n-1);

cout << "Отсортированный массив имеет вид: " << endl;

output(array, n);

cout << endl << endl;

break;

case 3:

QuickSort(array, n, 0, n-1);

cout << "Отсортированный массив имеет вид: " << endl;

output(array, n);

cout << endl << endl;

break;

}

} while (input != 4);

}

int \*\*init(int &n)

{

cout<<"Введите количество строк/столбиков в массиве: "<<endl;

cin>>n;

int \*\*array = new int\*[n];

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

array[i] = new int[n];

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

{

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

array[i][j] = rand() % 9;

}

return array;

}

void output(int \*\*array, int n)

{

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

{

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

{

cout<<array[i][j]<<" ";

}

cout<<endl;

}

}

void SortShell(int \*\*array, int n)

{

for (int k = 0; k < n; k++) {

int step = n / 2;

while (step > 0)

{

for (int j = 0; j < (n - step); j ++)

{

int i = j;

while (i >= 0 && array[i][k] < array[i+step][k])

{

int tmp = array[i][k];

array[i][k] = array[i+step][k];

array[i+step][k] = tmp;

i -= step;

}

}

step --;

}

}

}

void SortHoora(int \*\*array, int n, int first, int last)

{

for (int k=0; k < n; k++) {

int j = first, i = last;

int step = -1;

int condition = 1;

if (first >= last)

return;

do

{

if (condition == (array[i][k] > array[j][k]))

{

swap(array[j][k], array[i][k]);

swap(j, i);

step = -step;

condition = !condition;

}

i += step;

} while (i != j);

SortHoora(array, n, first, j - 1);

SortHoora(array, n, j + 1, last);

}

}

void QuickSort(int \*\*array, int n, int first, int last)

{

for (int k = 0; k < n; k++) {

int x = array[(first + last) / 2][k];

int j = first;

int i = last;

while (j <= i)

{

while (array[j][k] > x) j++;

while (array[i][k] < x) i--;

if (j <= i)

{

if (array[j][k] < array[i][k])

swap(array[j][k], array[i][k]);

j++;

i--;

}

}

if (j < last) {

QuickSort(array, n, j, last);

}

if (first < i) {

QuickSort(array, n, first, i);

}

}

}