**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(МИИТ)**

|  |
| --- |
| **ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Кафедра «Управление и защита информации»

**Лабораторная работа №9**

**«**Реализация алгоритма быстрой сортировки**»**

по дисциплине

**«Методы программирования»**

**Выполнил:** ст. гр. ТКИ-311,

Виноградова В.Л.

**Проверил:**

к.т.н., доц. Логинова Л.Н.,

к.т.н., доц. Сафронов А.И.

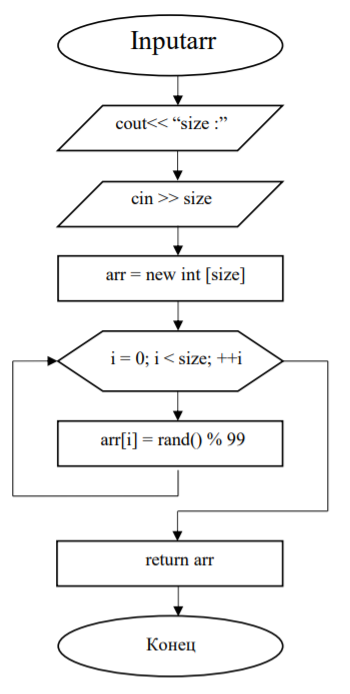
**Москва – 2022 г.**

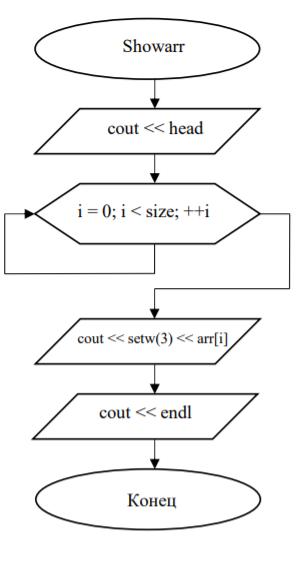
1. Цель работы

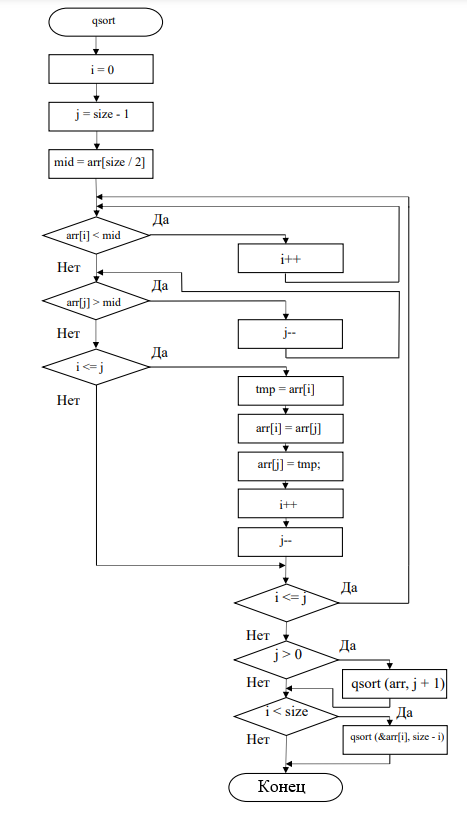
Изучить алгоритм быстрой сортировки.

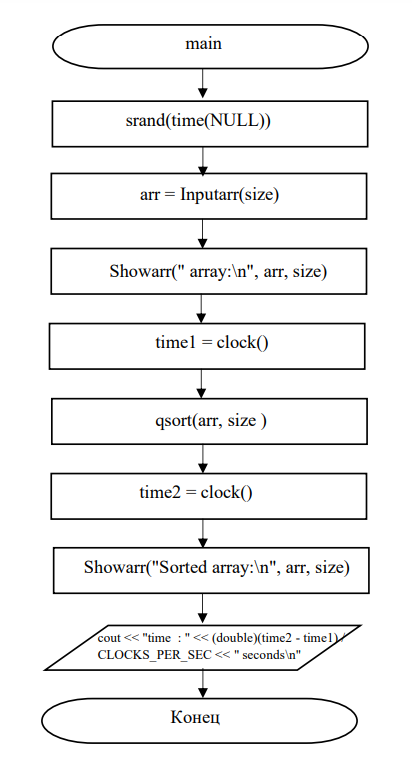
1. Постановка задачи

Реализовать алгоритм быстрой сортировки на любом языке программирования, ориентируясь на псевдокод. Сделать вывод о проделанной работе.

1. Блок-схема







1. Тестовый пример

Нужно отсортировать массив из 9 элементов:

Изображение выглядит как стол

Автоматически созданное описание

1. Код программы на языке С++

#include <iostream>

#include <ctime>

#include <iomanip>

using namespace std;

int\* Inputarr(int& size)

{

int i, \* arr;

cout << "array size: ";

cin >> size;

arr = new int[size];

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

arr[i] = rand() % 99;

return arr;

}

void Showarr(const char\* head, int\* arr, int size)

{

int i;

cout << head;

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

cout << setw(3) << arr[i];

cout << endl;

}

void qsort(int\* arr, int size) {

//Указатели в начало и в конец массива

int i = 0;

int j = size - 1;

//Центральный элемент массива

int mid = arr[size / 2];

//Делим массив на левую и правую части

do {

//Пробегаем элементы, ищем те, которые нужно перекинуть в другую часть

//В левой части массива пропускаем(оставляем на месте) элементы, которые меньше центрального

while (arr[i] < mid) {

i++;

}

//В правой части пропускаем элементы, которые больше центрального

while (arr[j] > mid) {

j--;

}

//Меняем элементы местами

if (i <= j) {

int tmp = arr[i];

arr[i] = arr[j];

arr[j] = tmp;

i++;

j--;

}

} while (i <= j);

//Рекурсивные вызовы, если осталось, что сортировать

if (j > 0) {

//"Левая часть "

qsort(arr, j + 1);

}

if (i < size) {

//"Првая часть"

qsort(&arr[i], size - i);

}

}

int main()

{

int\* arr, size;

clock\_t time1, time2;

srand(time(NULL));

arr = Inputarr(size);

Showarr(" array:\n", arr, size);

time1 = clock();

qsort(arr, size);

time2 = clock();

Showarr("Sorted array:\n", arr, size);

cout << "time : " << (double)(time2 - time1) / CLOCKS\_PER\_SEC << " seconds\n";

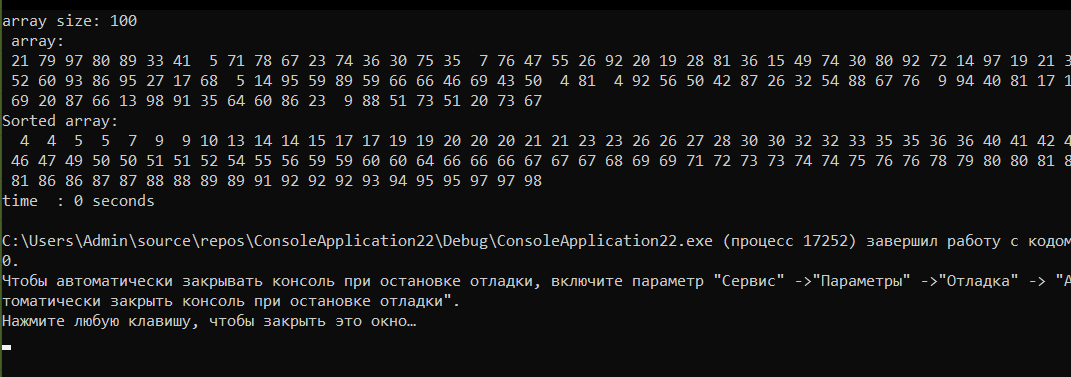
}

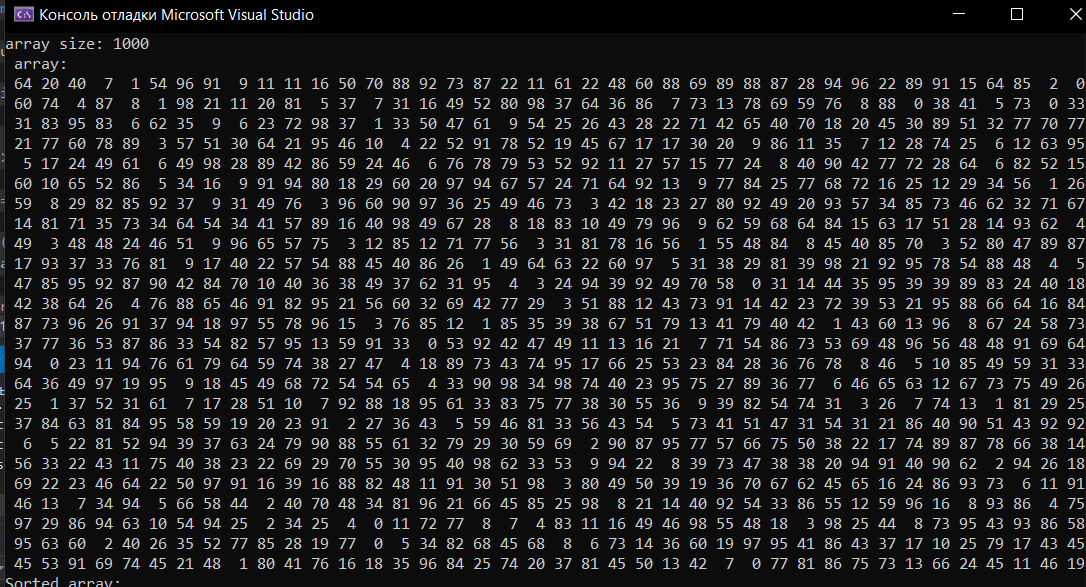
1. Результаты программ

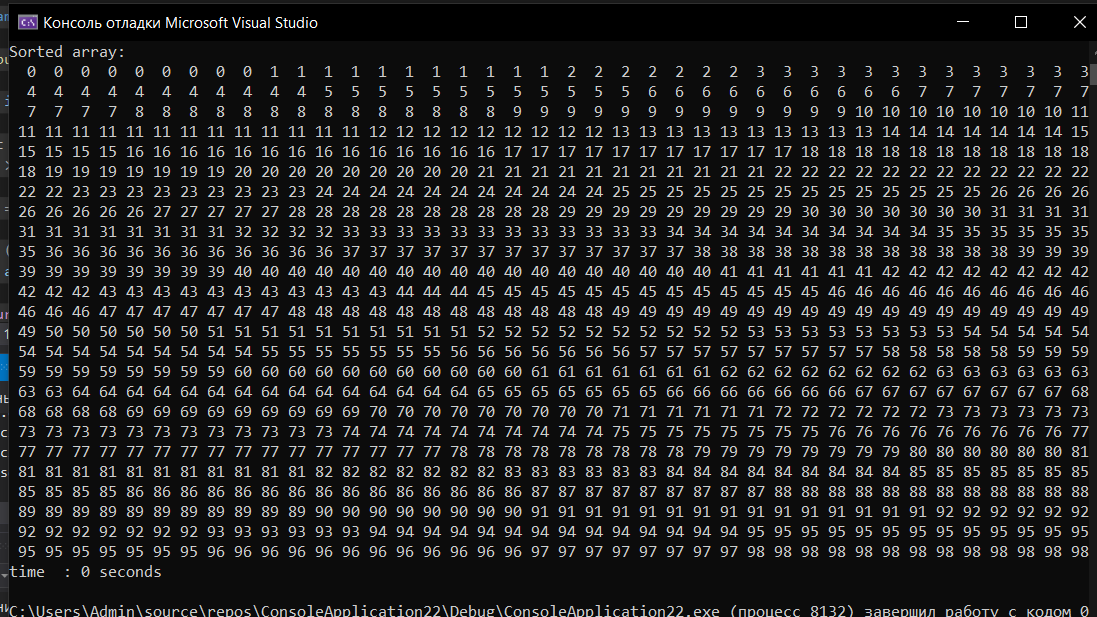
9 элементов:

Изображение выглядит как текст

Автоматически созданное описание

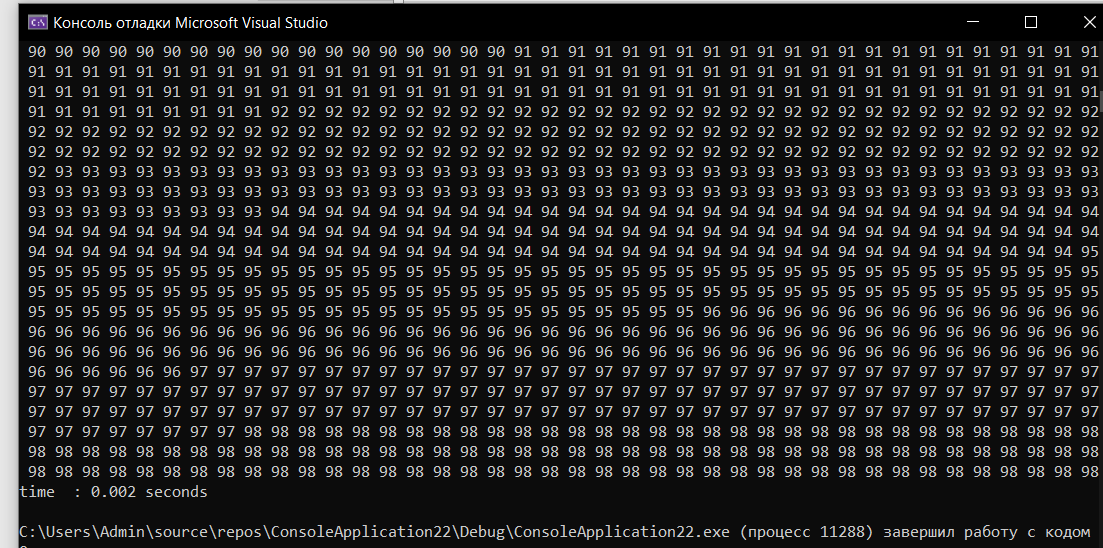
100 элементов:

1000 элементов:



Изображение выглядит как текст, электроника, компьютер

Автоматически созданное описание10 000 элементов:



1. Вывод

В ходе выполнения лабораторной работы был изучен и реализован алгоритм быстрой сортировки (Quicksort) массива. Программа работает правильно, работа выполнена в полном объеме.