МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра Информационная безопасность вычислительных систем и сетей

Методы сортировки

(наименование работы)

ОТЧЕТ

по лабораторной работе №5

по дисциплине

Технологии программирования

(наименование дисциплины)

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Капранов С.Н.

(подпись) (фамилия, и., о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Савельев М.А.

(подпись) (фамилия, и., о.)

18-ИСТ-4

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород, 2020

Содержание

[Задача 3](#_Toc35190540)

[Основная часть отчета 4](#_Toc35190541)

[Листинг программы 4](#_Toc35190542)

[Входные и выходные данные 7](#_Toc35190543)

# Задача

**16 вариант:** Реализовать «быструю» сортировку

# Основная часть отчета

Программа написана на языке С++в среде разработки VisualStudio 2017.

## Листинг программы

***Quick\_sort.cpp***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <iostream>

#include "quick\_sort.h"

using namespace std;

//--- Вывод на экран ---\\

void print(int arr[], int n)

{

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

{

cout << arr[i] << "|";

}

cout << endl;

}

//--- ---\\

int main()

{

int n; // размер массива

cout << "Array Size: ";

//--- Обработка ввода ---\\\

while (!(cin >> n) || n < 0)

{

cout << "Uncorrect size, try again \n";

cout << "Array Size: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

cout << endl;

//--- ---\\\

int\* arr = new int[n]; // массив размером n

//--- Ввод n элементов массив поочереди ---\\

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

{

cout << "Array[" << i + 1 << "]: ";

while (!(cin >> arr[i]))

{

cout << "Uncorrect number, try again \n";

cout << "Array[" << i + 1 << "]: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

cout << endl;

}

//--- ---\\

print(arr, n); // вывод на экран введенной последовательности

quickSort(arr, 0, n - 1); // функция сортировки

print(arr, n); // вывод на экран результата сортировки

system("pause");

}

***Quick\_sort.h***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
#include <iostream>

using namespace std;

//--- Функция "быстрой" сортировки ---\\

void quickSort(int arr[], int left, int right)

{

int i = left, j = right; // границы массива

int tmp; // переменная, временно хранящая в себе значения

int pivot = arr[(left + right) / 2]; // нахождение середины массива

while (i <= j) { // пока не середина

while (arr[i] < pivot) // пока значения слева меньше серединного

i++;

while (arr[j] > pivot) // пока значения справа больше серединного

j--;

if (i <= j)

{

//--- Свап значений ---\\

tmp = arr[i];

arr[i] = arr[j];

arr[j] = tmp;

//--- ---\\

i++;

j--;

}

}

if (left < j)

quickSort(arr, left, j); // вызов функции для левой части массива

if (i < right)

quickSort(arr, i, right); // вызов функции для правой части массива

}

//--- ---\\

## Входные и выходные данные

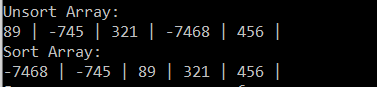
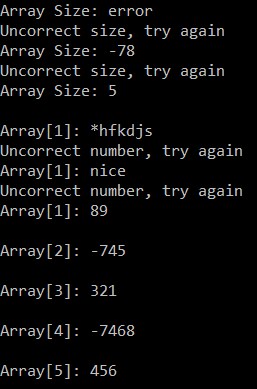


Рис. 1. Входные данные Рис. 2. Выходные данные