**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Кафедра безопасности информационных систем**

**ОТЧЁТ**

по практической работе работе №4 на тему:   
**«Сортировка числовых массивов. Некоторые методы сортировки»**

по дисциплине «Алгоритмы и структуры данных»

Выполнил: студент группы ИСТ-114,Константинов А.Д.

«29» октября 2022 г. \_\_\_\_\_\_\_\_\_\_\_/А.Д. Константинов/

Принял: к.ф.-м.н., доцент, И.А. Моисеев

« » ноября 2022 г. \_\_\_\_\_\_\_\_\_\_\_/ И.А. Моисеев /

**Содержание:**

[Цель работы: 3](#_Toc114692552)

[Листинг кода: 3](#_Toc114692553)

[Результат работы программы: 8](#_Toc114692554)

[Вывод: 8](#_Toc114692555)

[Приложение: 22](#_Toc114692556)

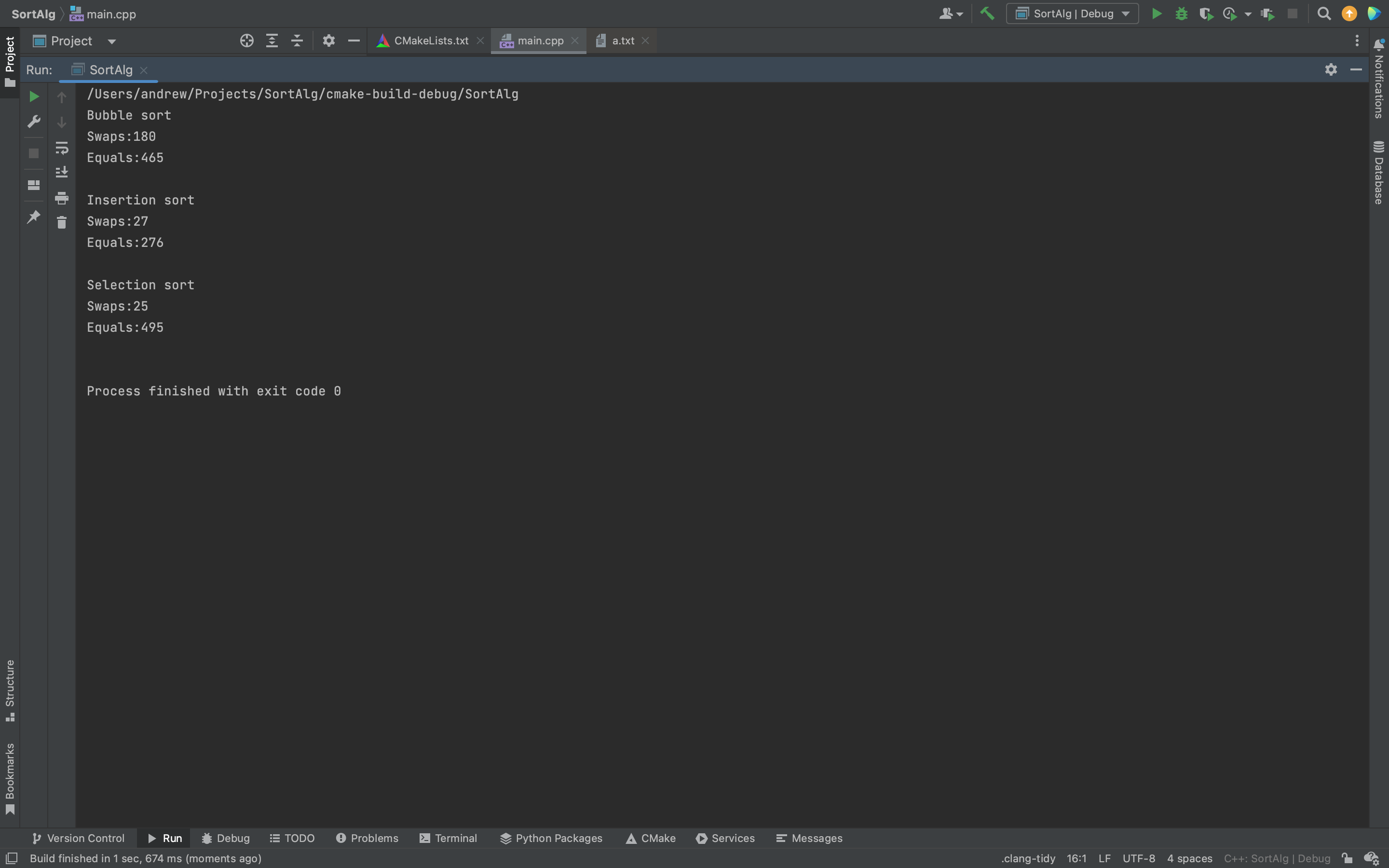
Цель работы:

# Требуется считать из файла, находящегося на диске одномерный массив, состоящий из 30 произвольных целых чисел и отсортировать его в порядке убывания приведенными выше методами. На экран вывести количество операций сравнения и пересылки чисел и время выполнения программы для каждого метода.

# Листинг кода:

#include <iostream>  
#include "fstream"  
#include "vector"  
int equals = 0;  
int swaps = 0;  
void SelectionSort(int Ndata[30])  
{  
 int data[30];  
 for (int i = 0; i < 30; ++i) {  
 data[i] = Ndata[i];  
 }  
 for (int i = 0; i < 30; ++i) {  
 int minInd = i;  
 for (int j = i; j < 30; ++j) {  
 equals++;  
 if(data[minInd] < data[j])  
 {  
 minInd = j;  
 }  
 }  
 equals++;  
 if(minInd!=i) {  
 swaps++;  
 int temp = data[minInd];  
 data[minInd] = data[i];  
 data[i] = temp;  
 }  
 }  
}  
void InsertionSort(int Ndata[30])  
{  
 int data[30];  
 for (int i = 0; i < 30; ++i) {  
 data[i] = Ndata[i];  
 }  
 std::vector<int> sorted;  
 sorted.push\_back(data[0]); // Массив из одного элемента по умолчанию отсортирован)  
 for (int i = 1; i < 30; ++i) {  
 ///Теперь смотрим куда вставить наш новый элемент  
 bool succ=false;  
 for (int j = 0; j < sorted.size(); ++j) {  
 equals++;  
 if(sorted[j] < data[i])  
 {  
 swaps++;  
 sorted.insert(sorted.begin()+j, data[i]);  
 succ = true;  
 break;  
 }  
 }  
 equals++;  
 if(!succ) sorted.push\_back(data[i]);  
 }  
}  
void BubbleSort(int Ndata[30])  
{  
 int data[30];  
 for (int i = 0; i < 30; ++i) {  
 data[i] = Ndata[i];  
 }  
 for (int i = 0; i < 30; ++i) {  
 for (int j = i; j < 30; ++j) {  
 equals++;  
 if(data[i] < data[j])  
 {  
 swaps++;  
 int tmp = data[i];  
 data[i] = data[j];  
 data[j] = tmp;  
 }  
 }  
 }  
}  
int main() {  
 int data[30];  
 std::fstream fs;  
 fs.open("a.txt",std::ios\_base::in);  
 for (int i = 0; i < 30; ++i) {  
 fs >> data[i];  
 }  
 fs.close();  
  
 BubbleSort(data);  
 std::cout << "Bubble sort" << std::endl << "Swaps:" << swaps << std::endl << "Equals:" << equals << std::endl << std::endl;  
 swaps = 0;  
 equals = 0;  
  
 InsertionSort(data);  
 std::cout << "Insertion sort" << std::endl << "Swaps:" << swaps << std::endl << "Equals:" << equals << std::endl << std::endl;  
 swaps = 0;  
 equals = 0;  
  
 SelectionSort(data);  
 std::cout << "Selection sort" << std::endl << "Swaps:" << swaps << std::endl << "Equals:" << equals << std::endl << std::endl;  
 swaps = 0;  
 equals = 0;  
 return 0;  
}

# Результат работы программы:



# Вывод:

В ходе выполнения практической работы была написана программа, сортирующая 30 чисел тремя различными алгоритмами сортировка(пузырек, вставка, выбором)