Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра **«**Информационные технологии и автоматизированные системы**»**

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Семестр 2

Тема: Внешние сортировки.

Выполнил работу

Студент группы РИС-22-1Б

Поважный В. Е.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

г. Пермь-2023

**Введение**

Для выполнения лабораторной работы требуется сделать отчет программы. Создание программ – отличный способ практики программирования.

**Постановка задачи**

Необходимо организовать сортировку чисел, используя многофазную сортировку.

**Код программы**

#include <iostream>

#include <fstream>

#include <string>

#include <stdio.h>

using namespace std;

int Series(string path1, string path2) {

ifstream readf(path1);

ofstream writef(path2);

string temp, cur;

int itemp, icur, counter = 0;

if (!readf.is\_open() || !writef.is\_open()) { cout << "Error" << endl; }

else {

readf >> cur;

writef << cur << "\n";

temp = cur;

while (true) {

readf >> cur;

if (readf.eof()) {

writef << "/";

counter++;

return counter;

}

itemp = stoi(temp);

icur = stoi(cur);

if (itemp <= icur) writef << cur << "\n";

else {

writef << "/" << "\n" << cur << "\n";

counter++;

}

temp = cur;

}

}

readf.close();

writef.close();

}

void Fibonacci(string path1, string path2, string path3, int counter) {

ifstream f2(path2);

ofstream f1(path1), f3(path3);

string cur;

int max\_series = 1, temp, min\_series = 1;

if (!f1.is\_open() || !f2.is\_open() || !f3.is\_open()) { cout << "Error" << endl; }

else {

while ((min\_series + max\_series) < counter) {

temp = max\_series;

max\_series += min\_series;

min\_series = temp;

}

f2 >> cur;

int left\_series = counter - min\_series, add\_series = (max\_series + min\_series) - counter;

while (min\_series > 0) {

while (cur != "/") {

f1 << cur << "\n";

f2 >> cur;

}

f1 << "/" << "\n";

f2 >> cur;

min\_series--;

}

while (left\_series > 0) {

while (cur != "/") {

f3 << cur << "\n";

f2 >> cur;

}

f3 << "/" << "\n";

f2 >> cur;

left\_series--;

}

for (int i = 0; i < add\_series; i++) {

f3 << "/" << "\n";

}

}

f1.close();

f2.close();

f3.close();

}

void MergeUntilEmpty(string path1, string path2, string path3) {

int num = Series(path1, path2);

if (num == 1) return;

Fibonacci(path1, path2, path3, num);

ifstream f1(path1), f3(path3);

ofstream f2(path2);

string temp1, temp2;

int itemp1, itemp2;

if (!f1.is\_open() || !f2.is\_open() || !f3.is\_open()) { cout << "Error" << endl; }

else {

while (!(f1.eof())) {

f1 >> temp1;

f3 >> temp2;

while (temp1 != "/" || temp2 != "/") {

if (temp1 == "/") {

f2 << temp2 << "\n";

f3 >> temp2;

}

else if (temp2 == "/") {

f2 << temp1 << "\n";

f1 >> temp1;

}

else {

itemp1 = stoi(temp1);

itemp2 = stoi(temp2);

if (itemp1 <= itemp2) {

f2 << temp1 << "\n";

f1 >> temp1;

}

else {

f2 << temp2 << "\n";

f3 >> temp2;

}

}

}

}

}

f1.close();

f3.close();

f2.close();

swap(path1, path2);

MergeUntilEmpty(path1, path2, path3);

}

int main() {

srand(time(NULL));

int size, current;

string firstfile = "1.txt", secondfile = "2.txt", thirdfile = "3.txt", temp;

ofstream fir;

ifstream res;

cout << "Size of an array: ";

cin >> size;

cout << "unsorted data" << endl;

fir.open(firstfile);

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

current = rand() % 100;

cout << current << " ";

fir << current << "\n";

}

fir.close();

cout << endl << endl;

MergeUntilEmpty(firstfile, secondfile, thirdfile);

cout << "sorted data" << endl;

res.open(firstfile);

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

res >> temp;

cout << temp << " ";

}

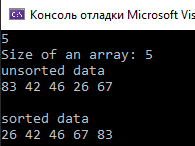
res.close();

cout << endl;

return 0;

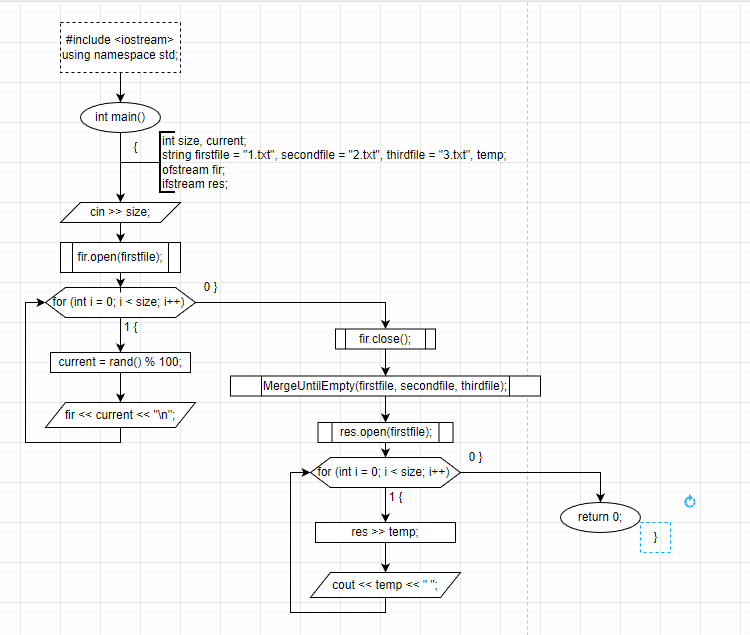
}

**Вывод программы**

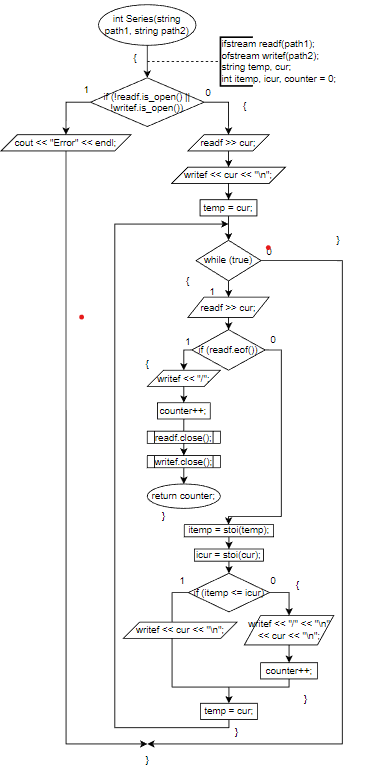
****

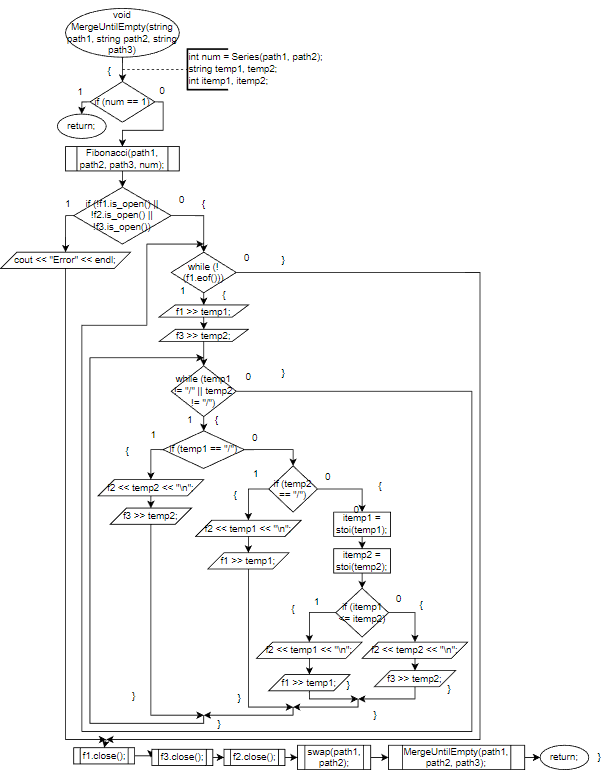
***Рисунок 1 – вывод программы***

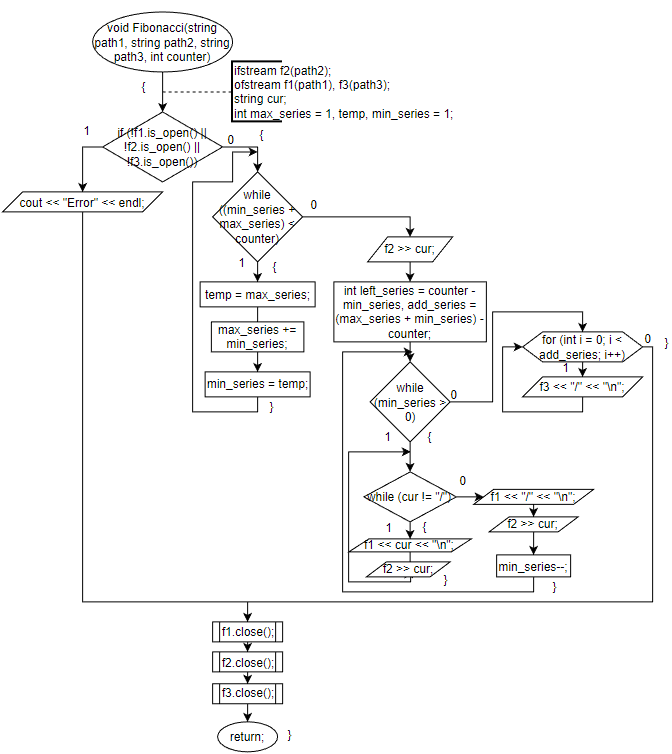
**Блок-схема**

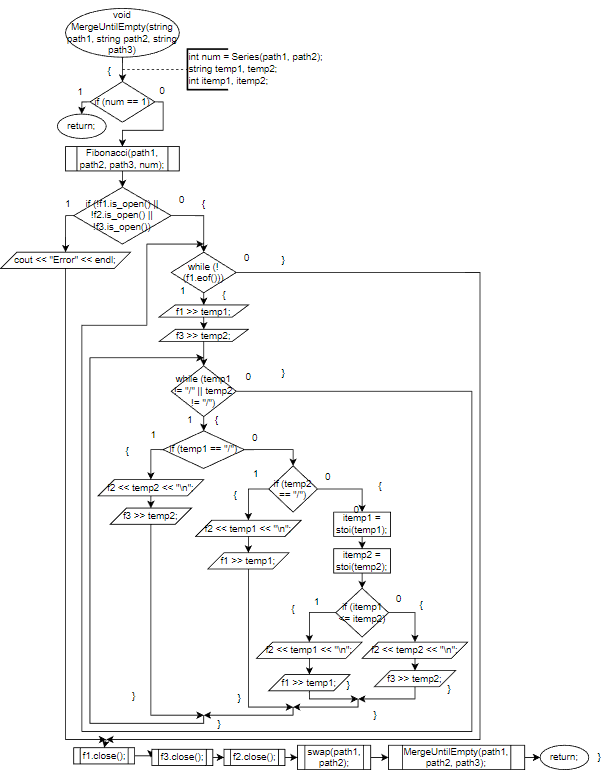
****

***Рисунок 3 – Схема алгоритма программы***

******

******





**Вывод**

Программа выполняет свою задачу.

**Заключение**

Для решения задачи потребовались знания языка программирования, а конкретнее C++. Программа выполняет те условия, что были указаны в постановке задачи и работает без проблем.