

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное образовательное
учреждение высшего образования
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ(МИИТ))**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**ОТЧЕТ ПО ДИСЦИПЛИНЕ
«АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»
ЛАБОРАТОРНАЯ РАБОТА №2**

Направление: 10.03.01 Информационная безопасность

Профиль: Безопасность компьютерных систем

Выполнил:
студент группы УИБ-115
Клепиков Степан Даниилович

Проверил:

(должность, ФИО)

(должность, ФИО)

Москва 2022 г.

Задание

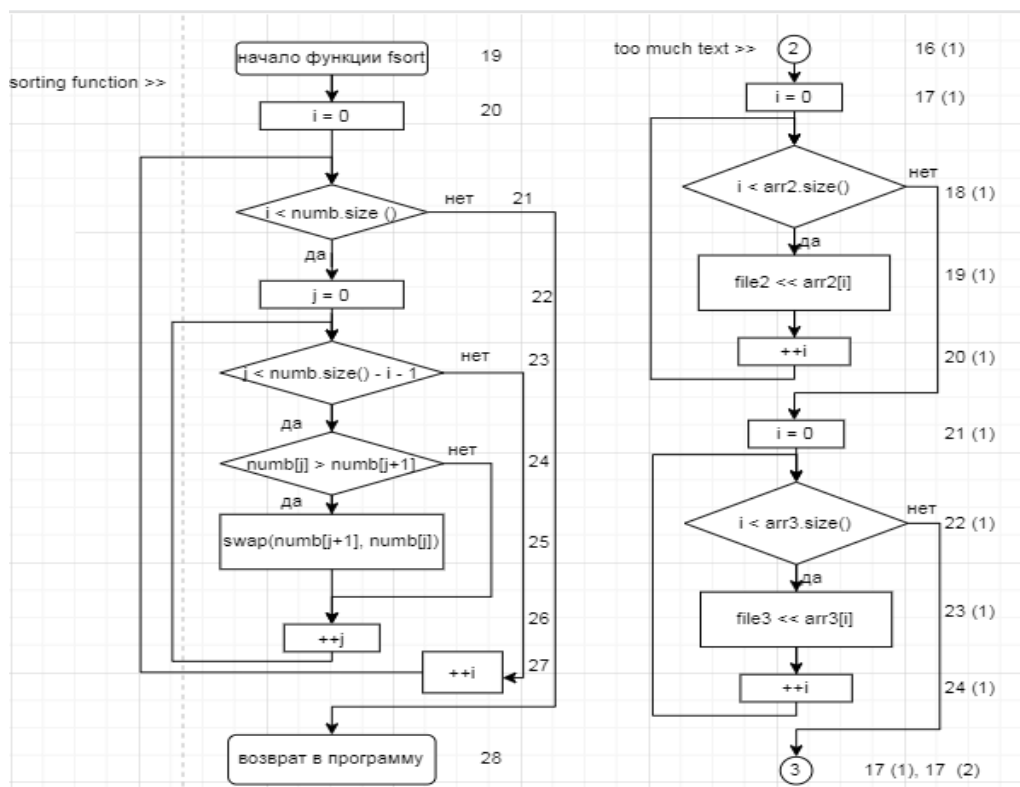
Задан файл f1, компоненты которого – целые числа. Записать в файл f2 все четные числа файла f1, а в файл f3 – все нечетные. Выполнить сортировку чисел в файлах f2 и f3 по возрастанию. Использовать функции, а также меню для выбора существующего или вновь создаваемого файла.

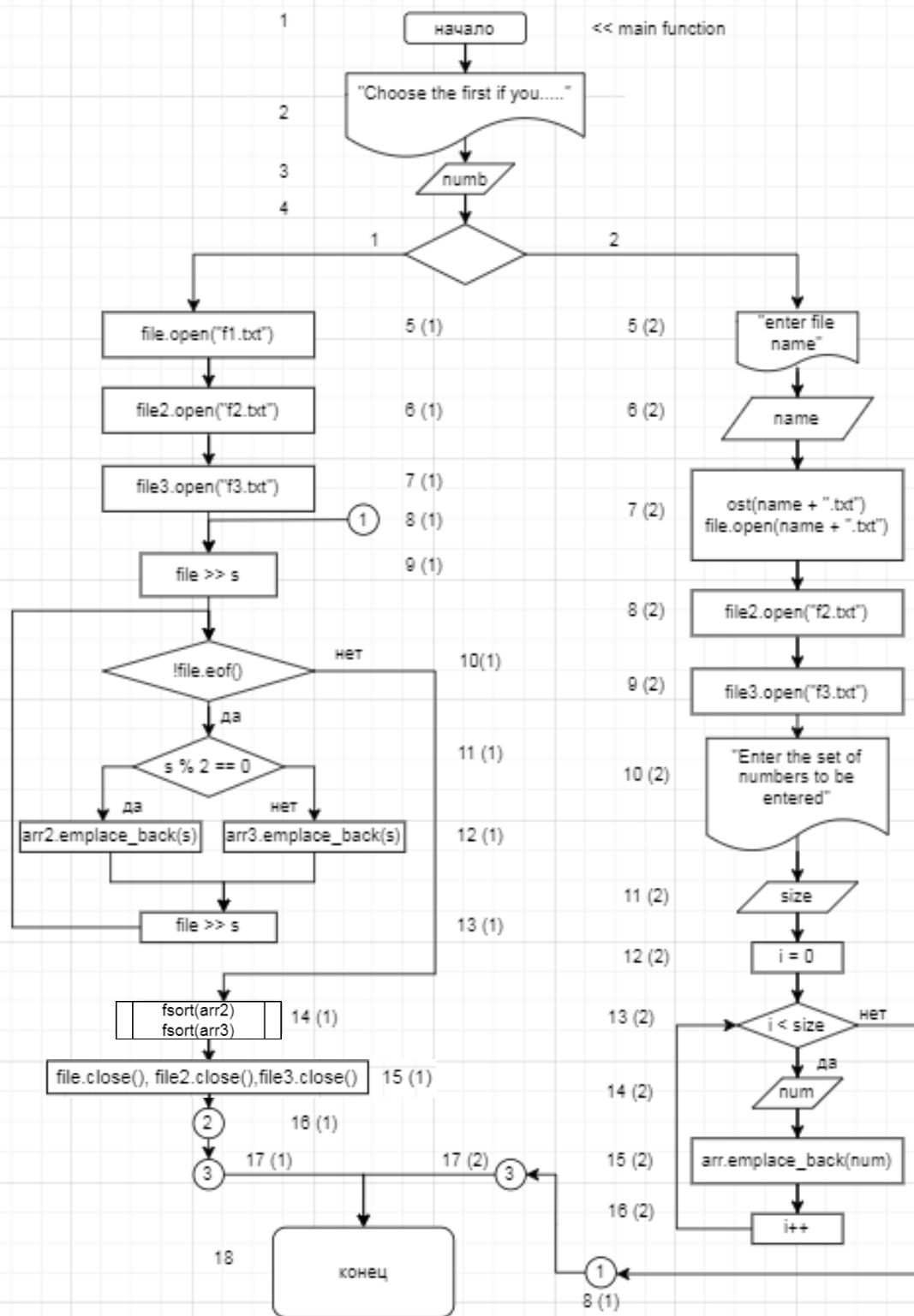
1. Таблица имён:

Рабочие переменные		
i	integer	Счетчик
j	integer	Счетчик
numb	integer	Переменная выбора варианта
s	integer	Символ файла
numb	vector	Временная переменная
name	string	Имя создаваемого файла
size	integer	Кол-во вводимых символов
num	integer	Вводимый символ
Исходные данные		
arr	integer	Исходный файл
Результат		
arr2	integer	Файл с четными числами
arr3	integer	Файл с нечетными числами

2. Блок-схема:

(внешняя и большая внутренняя функции)





3. Отладочный пример:

1. Начало
2. Вывод 'Choose the first if you want to take a ready-made, and the second if you want to create'
3. Ввод числа (1)
4. Проверка числа (1 => левая часть блок-схемы, (1))
5. (1) Открытие первого файла (13245678910)
6. (1) Открытие второго файла
7. (1) Открытие третьего файла
8. (1) Начало функции, используемой в правой части БС
9. (1) Ввод символа из file
10. (1) Проверка условия (!file.eof()) (верно)
11. (1) Проверка символа на четность (1%2 != 0 => неверно)
12. (1) Запись символа (1) в arr3
13. (1) Сдвиг на символ вперед (2)
 - 10 (1) Проверка условия (!file.eof()) (верно)
 - 11 (1) Проверка символа на четность (2%2 == 0 => верно)
 - 12 (1) Запись символа (2) в arr2
 - 13 (1) Сдвиг на символ вперед (3)
-
- 10 (1) Проверка условия (!file.eof()) (неверно)
- 14.(1) Вызов внешней функции fsort для arr2 и arr3
15. (1) Заккрытие открытых файлов
16. (1) Переход к объемной функции
17. (1) выход из последней функции
18. Завершение работы программы

Возврат к 4 пункту, выбирая 2

1. Начало
2. Вывод...
3. Ввод числа (2)
4. Проверка числа (2 => правая часть блок-схемы, (2))
5. (2) Вывод "enter file name"
6. (2) Ввод имени файла
7. (2) Создание первого файла с заданным (6(2)) именем
8. (2) Открытие второго файла
9. (2) Открытие третьего файла
10. (2) Вывод "Enter the set of numbers to be entered"
11. (2) Ввод количества символов (size = 4)
12. (2) i=0
13. (2) Проверка условия (i < size) (0 < 4 => верно)
14. (2) Ввод символа (1)
15. (2) Запись символа в arr

16. (2) Присвоение переменной i значения $i+1$ (1)
- 13 (2) Проверка условия ($i < \text{size}$) ($1 < 4 \Rightarrow$ верно)
-
- 13 (2) Проверка условия ($i < \text{size}$) ($5 < 4 \Rightarrow$ неверно)
- 8(1) Начало функции из левой части блок-схемы
- 17.(2) Возврат из объемной функции
18. Завершение работы программы

Внешняя и большая внутренняя функции

- 16.(1) Переход к данной функции из основной блок-схемы
- 17.(1) $i=0$
18. (1) Проверка условия ($i < \text{arr2.size}()$)
19. (1) Запись символа в `file2`
20. (1) Присвоение переменной i значения $i+1$
21. (1) $i=0$
22. (1) Проверка условия ($i < \text{arr3.size}()$)
23. (1) Запись символа в `file3`
24. (1) Присвоение переменной i значения $i+1$
- 17(1), 17(2) Выход из данной функции

Внешняя функция `fsort`

19. Начало функции `fsort`
20. $i=0$
21. Проверка условия ($i < \text{numb.size}()$)
- ($0 < 10$ в первом и 4 во втором случае. Рассматривается первый случай)
22. $j=0$
23. Проверка условия ($j < \text{numb.size}() - i - 1$) ($0 < 10 - 0 - 1$)
24. Проверка условия ($\text{numb}[j] > \text{numb}[j+1]$) ($1 > 2 \Rightarrow$ неверно)
26. $j++$
 - 23 Проверка условия ($j < \text{numb.size}() - i - 1$) ($1 < 10 - 0 - 1$)
 - 24 Проверка условия ($\text{numb}[j] > \text{numb}[j+1]$) ($3 > 2 \Rightarrow$ верно)
 - 25 Смена символов местами
 - 26 $j++$
 -
27. $i++$
 -
28. Возврат в программу

4. Код программы:

```
#include <iostream> //подключение библиотек
#include <string>
#include <vector>
#include <fstream>
using namespace std;

void fsort(vector <int>& numb) { //алгоритм сортировки по возрастанию
    for (int i = 0; i < numb.size(); ++i) {
        for (int j = 0; j < numb.size() - i - 1; ++j) {
            if (numb[j] > numb[j + 1])
                swap(numb[j + 1], numb[j]);
        }
    }
}

int main() { //основная программа
    int numb; //объявление переменных различных типов
    cout << "Choose the first if you want to take a ready-made, and the second if you want to
create" << endl; //ВЫВОД УСЛОВИЙ
    cin >> numb; //ВВОД номера условия

    switch (numb) //условие
    {
    case 1: //первый случай
    {
        int s;
        ifstream file; // файловый ввод-вывод
        ofstream file2;
        ofstream file3;
        vector <int> arr; //создание одноименных массивов
        vector <int> arr2;
```

```

vector <int> arr3;

file.open("f1.txt");//открытие файлов
file2.open("f2.txt");
file3.open("f3.txt");
for (file >> s; !file.eof(); file >> s)
{
    if (s % 2 == 0) {//проверка на четность
        arr2.emplace_back(s);
    }
    else
    {
        arr3.emplace_back(s);
    }
}

fsort(arr2); //вызов функции fsort
fsort(arr3);

for (int i : arr2)
{
    file2 << i << endl; //вывод символов в file3
}
for (int i : arr3)
{
    file3 << i << endl; //вывод символов в file3
}

file.close();//закрытие файлов
file2.close();
file3.close();
}

case 2:
{
    string name; //объявление переменных различных типов

```

```

ifstream file; // файловый ввод-вывод

ofstream file2;

ofstream file3;

int s, size, num;

vector<int> arr; //создание одноименных массивов

vector<int> arr2;

vector<int> arr3;

cout << "enter file name" << endl;

cin >> name; //ввод имени файла

ofstream ost(name + ".txt");

file.open(name + ".txt");//открытие файлов

file2.open("f2.txt");

file3.open("f3.txt");


cout << "Enter the set of numbers to be entered" << endl;


cin >> size; //ввод количества вводимых чисел


for (int i = 0; i < size; i++)
{
    cin >> num; //ввод символов

    arr.emplace_back(num); //заполнение вектора числами
}


for (file >> s; !file.eof(); file >> s)
{
    if (s % 2 == 0) { //проверка на четность

        arr2.emplace_back(s);

    }

    else

    {

        arr3.emplace_back(s);
    }
}

```



```

    }
}

fsort(arr2); //вызов функции fsort
fsort(arr3);

for (int i : arr2)
{
    file2 << i << endl; //ВЫВОД СИМВОЛОВ в file2
}
for (int i : arr3)
{
    file3 << i << endl; //ВЫВОД СИМВОЛОВ в file3
}

file.close();//закрытие файлов
file2.close();
file3.close();
}
}

return 0;
}

```

5. Результат выполнения работы программы:

1	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

1	2	1	1
2	4	2	3
3	6	3	5
4	8	4	7
5		5	9

6. Вывод:

В ходе выполнения работы были изучены базовые алгоритмы и функции языка высокого уровня C++.

Был произведен анализ их работы, на основании которого была составлена блок-схема. На контрольных примерах было выяснено, что все написанные функции работают корректно и в коде отсутствуют ошибки.

Был оформлен комплект документации на программный код.