

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Пермский национальный исследовательский политехнический
университет
Электротехнический факультет
Кафедра информационных технологий и автоматизированных систем

Лабораторная работа
"Внешние сортировки"
Вариант: 12

Выполнил студент ИВТ-24-26:
Шишкин Максим Григорьевич

(дата, подпись)

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

(дата, подпись)

Пермь 2025

Содержание

1 Постановка задачи.....	3
2 Естественное слияние	4
2.1 Код на с++.....	4-6
2.2 Результат работы программы	6
2.3 Блок-схема	7-8
3 Многофазная	8
3.1 Код на с++.....	8-10
3.2 Результат работы программы	10
3.3 Блок-схема.....	11-13
4 Сбалансированное слияние.....	14
4.1 Код на с++.....	14-15
4.2 Результат работы программы	15
4.3 Блок-схема.....	16-18
5 Ссылка на github.....	18

1 Постановка задачи

Отсортировать массив чисел в файле используя три метода внешней сортировки: метод естественного слияния, сбалансированного слияния и многофазная сортировка.

2 Метод естественного слияния

2.1 Код на C++

```
1  ✓ #include <iostream>
2  | #include <fstream>
3  | #include <locale>
4  | using namespace std;
5
6  ✓ void split(const char* source, const char* file1, const char* file2) {
7  |     ifstream in(source);
8  |     ofstream out1(file1);
9  |     ofstream out2(file2);
10
11 |     int x, prev;
12 |     bool toFirst = true;
13
14 |     if (in >> prev) {
15 |         while (in >> x) {
16 |             if (toFirst) {
17 |                 out1 << prev << " ";
18 |             }
19 |             else {
20 |                 out2 << prev << " ";
21 |             }
22
23 |             if (x < prev) {
24 |                 toFirst = !toFirst;
25 |             }
26 |             prev = x;
27 |         }
28 |         if (toFirst) {
29 |             out1 << prev << " ";
30 |         }
31 |         else {
32 |             out2 << prev << " ";
33 |         }
34 |     }
35 |
36 |     in.close();
37 |     out1.close();
38 |     out2.close();
39 | }
40
41 ✓ void merge(const char* file1, const char* file2, const char* dest) {
42 |     ifstream in1(file1);
```

```

43     ifstream in2(file2);
44     ofstream out(dest);
45
46     int a, b;
47     bool hasA = false, hasB = false;
48
49     if (in1 >> a) hasA = true;
50     if (in2 >> b) hasB = true;
51
52     while (hasA && hasB)
53     {
54         if (a <= b) {
55             out << a << " ";
56             if (in1 >> a)
57                 hasA = true;
58             else
59                 hasA = false;
60         }
61         else {
62             out << b << " ";
63             if (in2 >> b)
64                 hasB = true;
65             else
66                 hasB = false;
67         }
68     }
69
70     while (hasA)
71     {
72         out << a << " ";
73         if (in1 >> a)
74             hasA = true;
75         else
76             hasA = false;
77     }
78
79     while (hasB)
80     {
81         out << b << " ";
82         if (in2 >> b)
83             hasB = true;
84         else

```

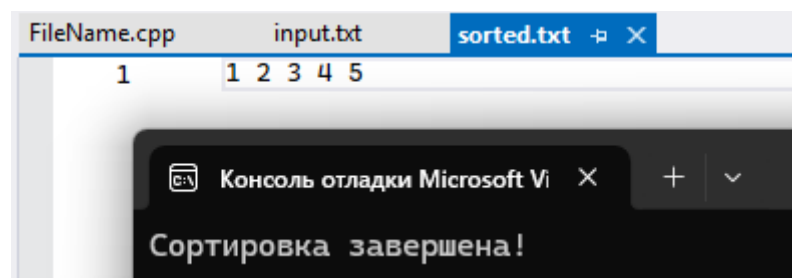
```

85         hasB = false;
86     }
87
88     in1.close();
89     in2.close();
90     out.close();
91 }
92
93 bool isSorted(const char* file) {
94     ifstream in(file);
95     int x, y;
96
97     if (!(in >> x)) {
98         in.close();
99         return true;
100     }
101
102     while (in >> y) {
103         if (y < x) {
104             in.close();
105             return false;
106         }
107         x = y;
108     }
109
110     in.close();
111     return true;
112 }
113
114 int main() {
115     setlocale(LC_ALL, "ru");
116     const char* input = "input.txt";
117     const char* output = "sorted.txt";
118     const char* temp1 = "temp1.txt";
119     const char* temp2 = "temp2.txt";
120
121     while (!isSorted(input)) {
122         split(input, temp1, temp2);
123         merge(temp1, temp2, input);
124     }
125
126     rename(input, output);
127
128     cout << "Сортировка завершена!" << endl;
129     return 0;
130 }

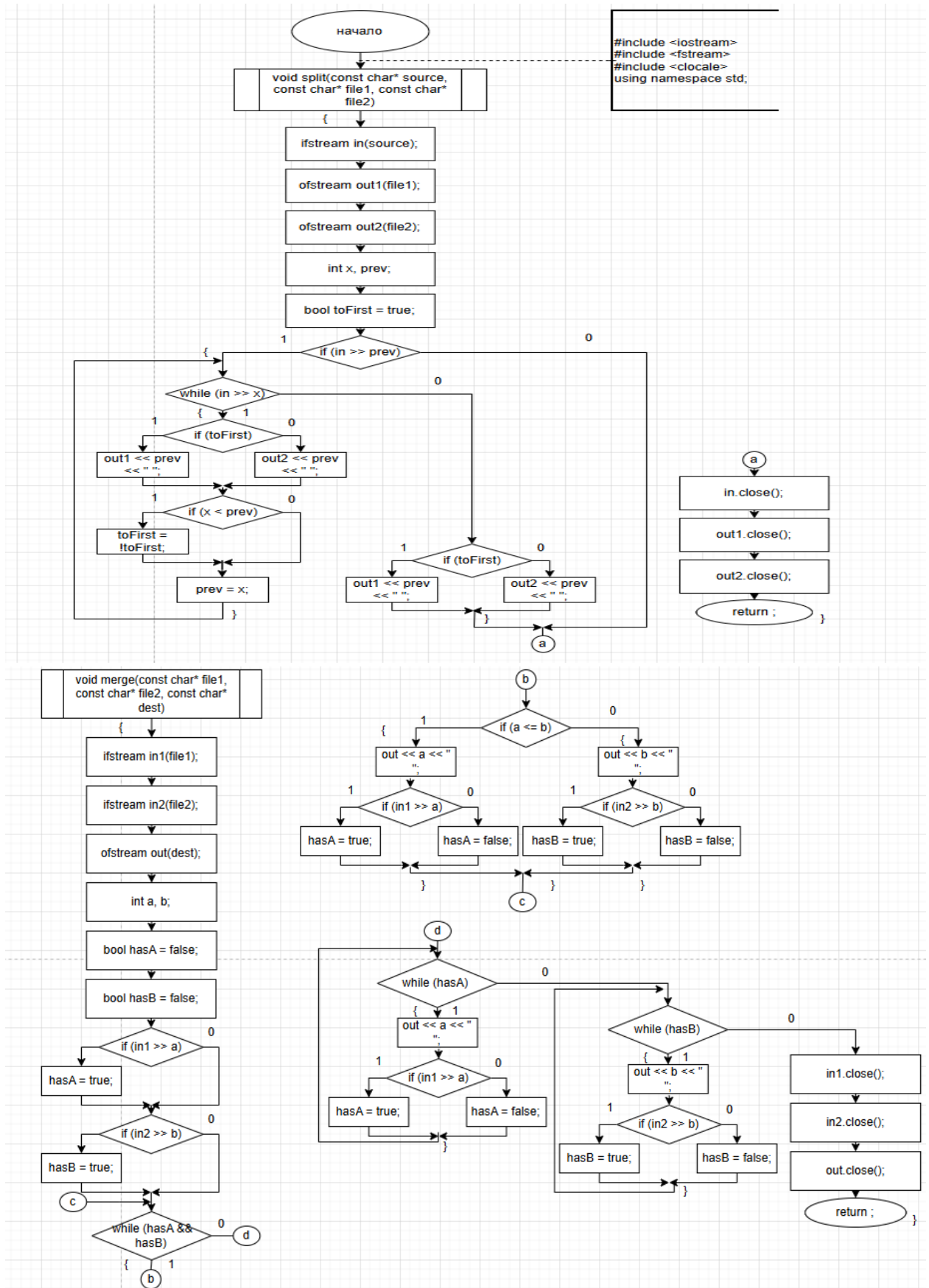
```

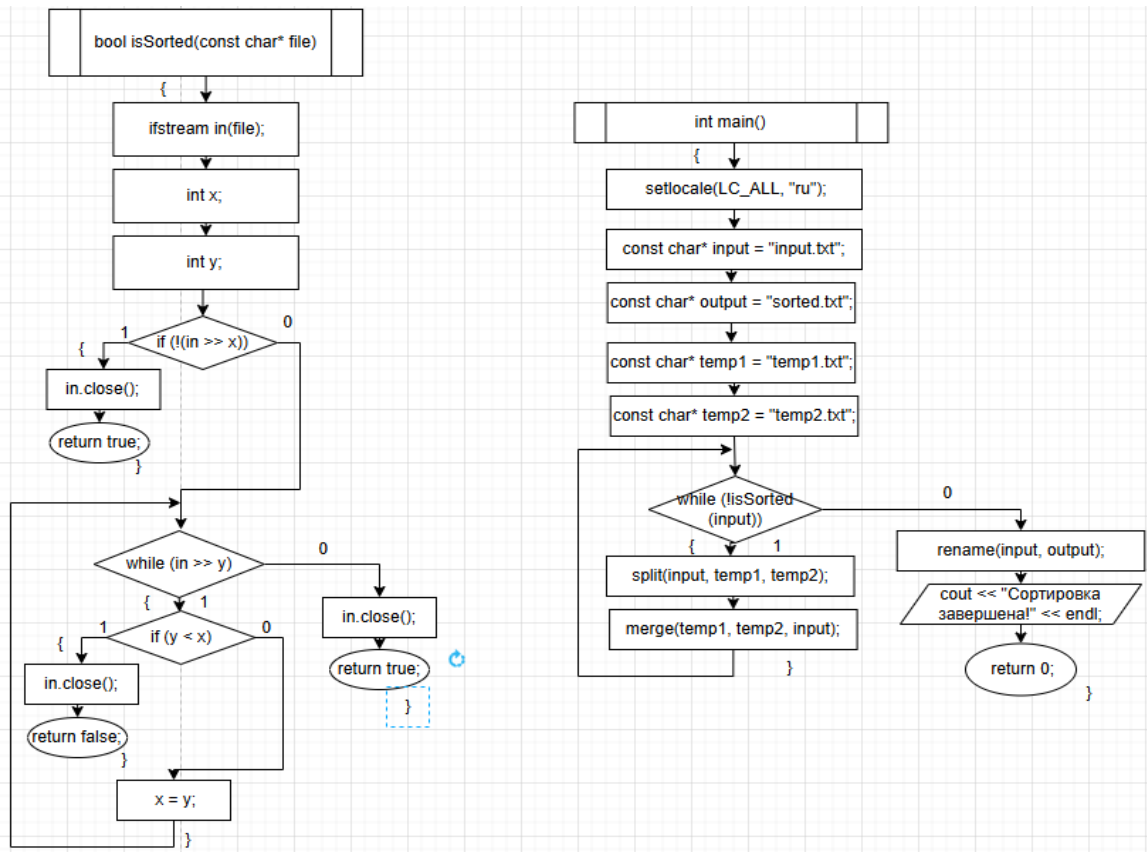
2.2 Результат работы

Исходный массив: 5 4 3 1 2



2.3 Блок-схема





3 Многофазная

3.1 Код на C++

```

1  #include <iostream>
2  #include <fstream>
3  #include <locale>
4  using namespace std;
5
6  const int CHUNK_SIZE = 100;
7
8  void sortChunk(int arr[], int size) {
9      for (int i = 0; i < size - 1; i++)
10         for (int j = 0; j < size - i - 1; j++)
11             if (arr[j] > arr[j + 1]) swap(arr[j], arr[j + 1]);
12 }
13
14 void splitToFiles(const string& input)
15 {
16     ifstream in(input);
17     ofstream f1("f1.txt"), f2("f2.txt"), f3("f3.txt");
18     int chunk[CHUNK_SIZE];
19     int count = 0;
20     int fileNum = 0;
21
22     while (in >> chunk[count])
23     {
24         count++;
25         if (count == CHUNK_SIZE)
26         {
27             sortChunk(chunk, count);
28
29             // Запись в соответствующий файл
30             if (fileNum == 0)
31             {
32                 for (int i = 0; i < count; i++) f1 << chunk[i] << " ";
33             }
34             else if (fileNum == 1)
35             {

```



```

36         for (int i = 0; i < count; i++) f2 << chunk[i] << " ";
37     }
38     else
39     {
40         for (int i = 0; i < count; i++) f3 << chunk[i] << " ";
41     }
42
43     count = 0;
44     fileNum = (fileNum + 1) % 3;
45 }
46
47
48 // Если остались данные, нужно их отсортировать и записать
49 if (count > 0) {
50     sortChunk(chunk, count);
51     if (fileNum == 0)
52     {
53         for (int i = 0; i < count; i++) f1 << chunk[i] << " ";
54     }
55     else if (fileNum == 1)
56     {
57         for (int i = 0; i < count; i++) f2 << chunk[i] << " ";
58     }
59     else
60     {
61         for (int i = 0; i < count; i++) f3 << chunk[i] << " ";
62     }
63 }
64 }
65
66 bool isEmpty(const string& filename) {
67     ifstream file(filename);
68     bool empty = true;
69     int temp;
70     if (file >> temp) {
71         empty = false;
72     }
73     file.close();
74     return empty;
75 }
76
77 void mergeFiles(const string& f1, const string& f2, const string& outFile) {
78     ifstream in1(f1), in2(f2);
79     ofstream out(outFile);
80     int a, b;
81     bool hasA = false, hasB = false;
82
83     if (in1 >> a) hasA = true;
84     if (in2 >> b) hasB = true;
85
86     while (hasA || hasB) {
87         if (!hasB) {
88             out << a << " ";
89             hasA = (in1 >> a) ? true : false;
90         }
91         else if (!hasA) {
92             out << b << " ";
93             hasB = (in2 >> b) ? true : false;
94         }
95         else if (a <= b) {
96             out << a << " ";
97             hasA = (in1 >> a) ? true : false;
98         }
99         else {
100             out << b << " ";
101             hasB = (in2 >> b) ? true : false;
102         }
103     }
104 }
105
106 void polyphaseSort(const string& input, const string& output) {
107     splitToFiles(input);
108
109     bool done = false;
110     while (!done) {
111         int nonEmpty = 0;
112         if (!isEmpty("f1.txt")) nonEmpty++;
113         if (!isEmpty("f2.txt")) nonEmpty++;
114         if (!isEmpty("f3.txt")) nonEmpty++;
115
116         if (nonEmpty <= 1) {
117             done = true;
118         }
119         else {

```

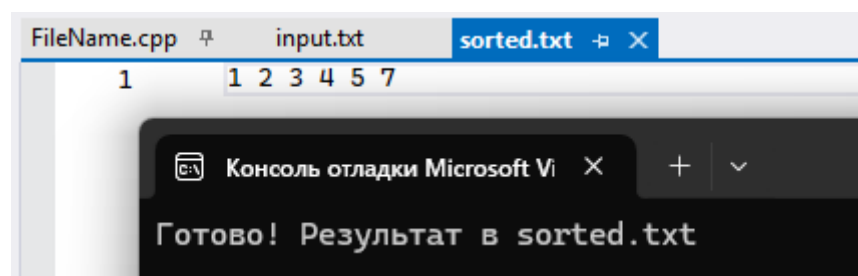
```

120     mergeFiles("f1.txt", "f2.txt", "temp.txt");
121
122     // Очищаем f1 и f2
123     ofstream("f1.txt").close();
124     ofstream("f2.txt").close();
125
126     // Переносим данные из temp в f1
127     ifstream temp("temp.txt");
128     ofstream f1("f1.txt");
129     int num;
130     while (temp >> num) f1 << num << " ";
131     temp.close();
132     f1.close();
133 }
134
135
136 // Определяем итоговый файл
137 ifstream resultFile;
138 ofstream outputFile(output);
139
140 if (!isFileEmpty("f1.txt")) {
141     resultFile.open("f1.txt");
142 }
143 else if (!isFileEmpty("f2.txt")) {
144     resultFile.open("f2.txt");
145 }
146 else {
147     resultFile.open("f3.txt");
148 }
149
150 int num;
151 while (resultFile >> num) {
152     outputFile << num << " ";
153 }
154
155 resultFile.close();
156 outputFile.close();
157
158 // Удаляем временные файлы
159 remove("temp.txt");
160 remove("f2.txt");
161 remove("f3.txt");
162 }
163
164 int main() {
165     setlocale(LC_ALL, "ru");
166     polyphaseSort("input.txt", "sorted.txt");
167     cout << "Готово! Результат в sorted.txt" << endl;
168     return 0;
169 }

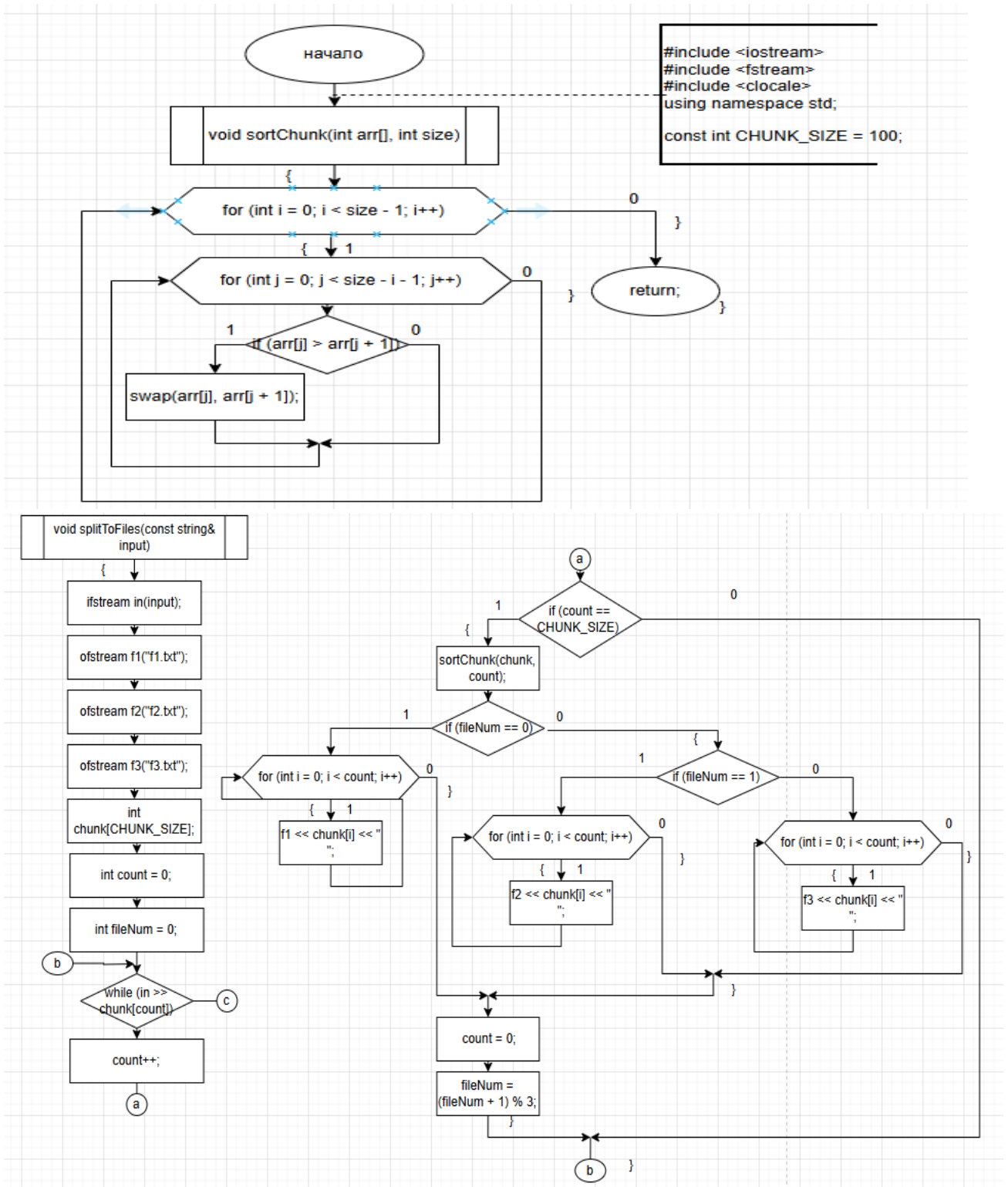
```

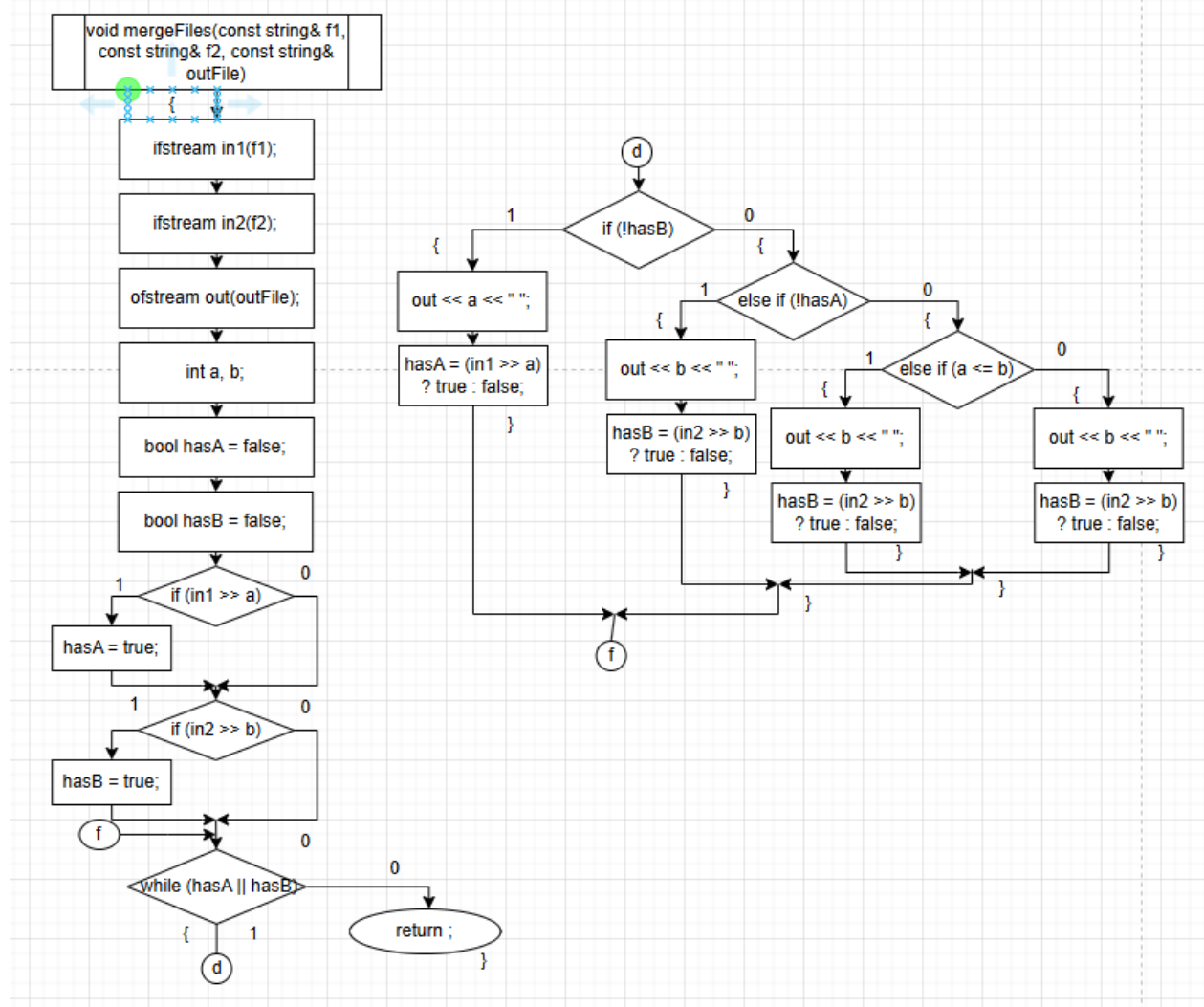
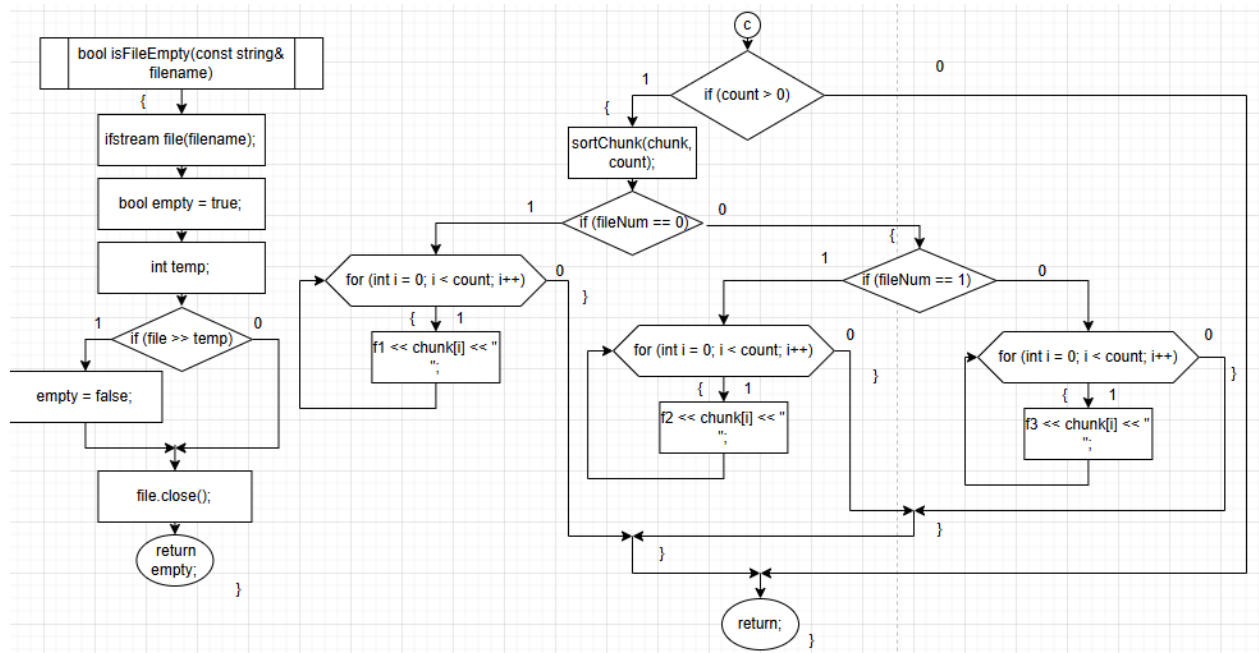
3.2 Результат работы

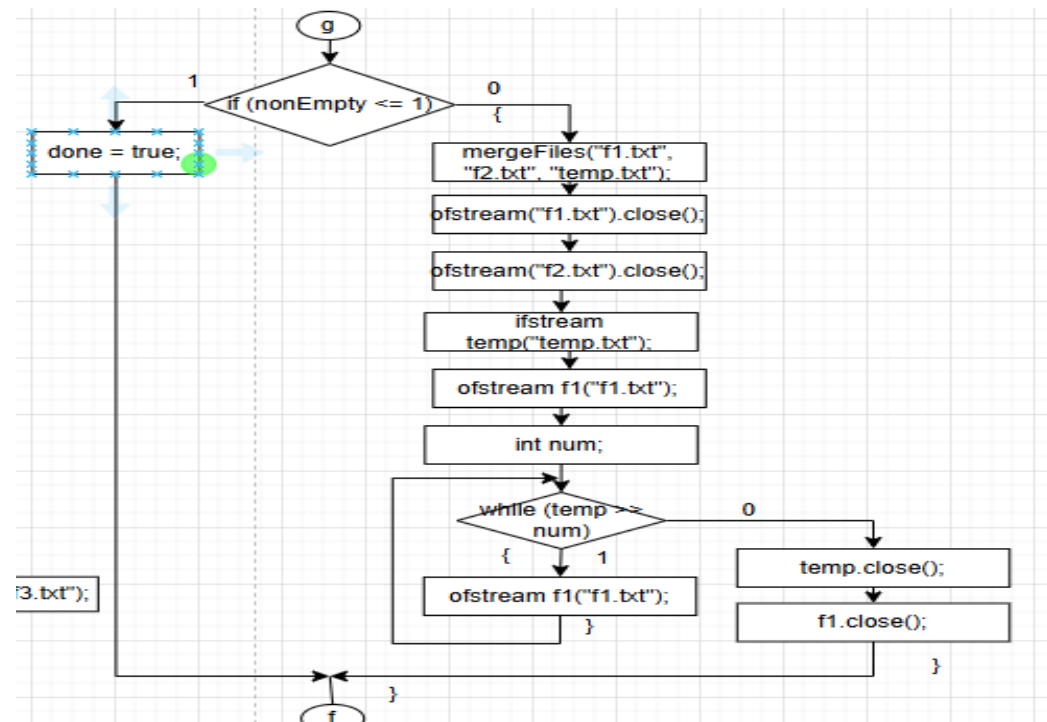
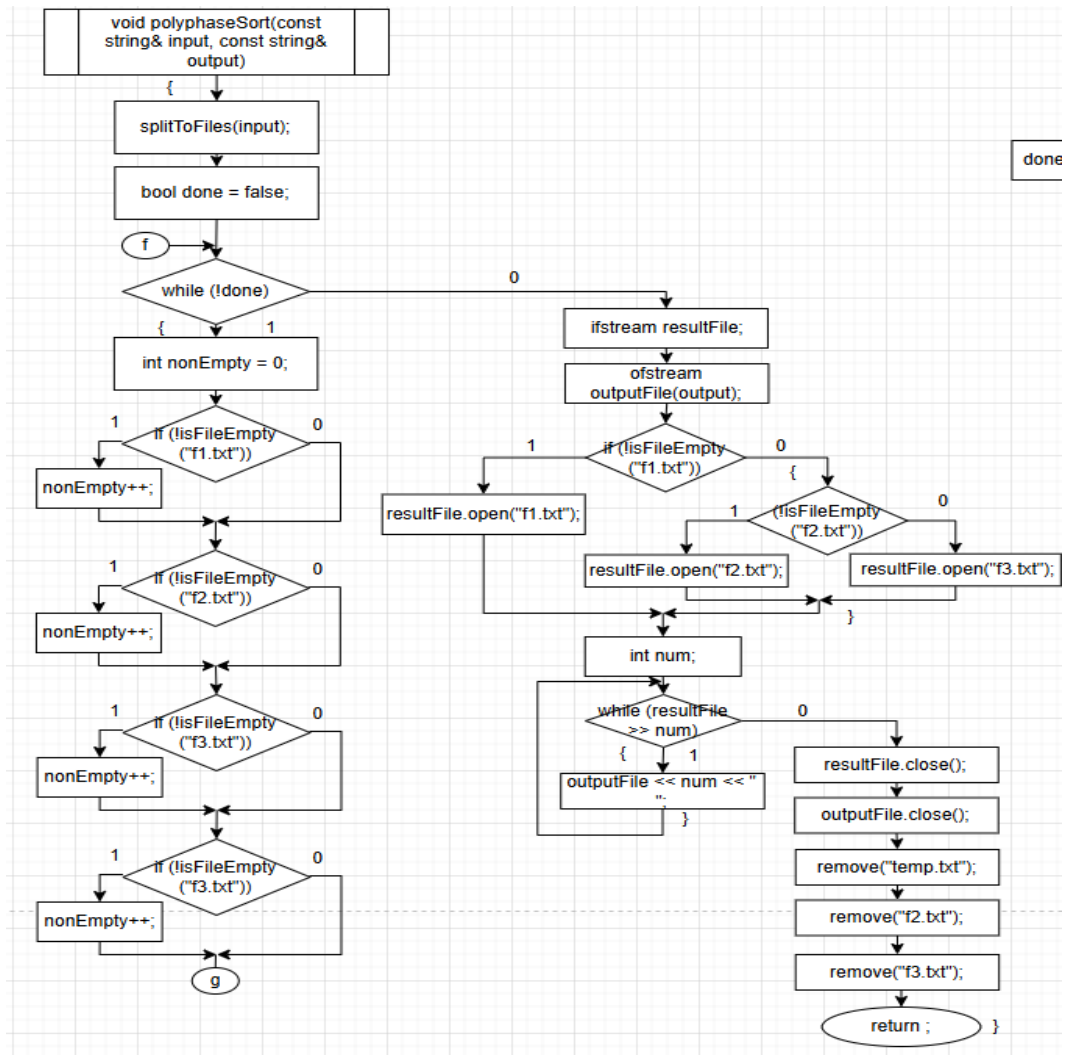
Исходный массив: 4 3 2 1 5 7



3.3 Блок-схема







4 Сбалансированное слияние

4.1 Код на C++

```
1  #include <iostream>
2  #include <fstream>
3  #include <locale>
4
5  using namespace std;
6
7  // Копирование блоков заданной длины
8  void split(const char* source, const char* file1, const char* file2, int blockSize) {
9      ifstream in(source);
10     ofstream out1(file1);
11     ofstream out2(file2);
12
13     int x;
14     bool toFirst = true;
15     int count = 0;
16
17     while (in >> x) {
18         if (toFirst) {
19             out1 << x << " ";
20         }
21         else {
22             out2 << x << " ";
23         }
24         count++;
25         if (count == blockSize) {
26             count = 0;
27             toFirst = !toFirst;
28         }
29     }
30
31     in.close();
32     out1.close();
33     out2.close();
34 }
35
36 // Слияние блоков фиксированной длины
37 void merge(const char* file1, const char* file2, const char* dest, int blockSize) {
38     ifstream in1(file1);
39     ifstream in2(file2);
40     ofstream out(dest);
41
42     int a, b;
43     bool hasA = false, hasB = false;
44
45     if (in1 >> a) hasA = true;
46     if (in2 >> b) hasB = true;
47
48     while (hasA || hasB)
49     {
50         int countA = 0, countB = 0;
51
52         while (countA < blockSize && countB < blockSize && hasA && hasB)
53         {
54             if (a <= b)
55             {
56                 out << a << " ";
57                 countA++;
58                 if (in1 >> a) hasA = true;
59                 else hasA = false;
60             }
61             else
62             {
63                 out << b << " ";
64                 countB++;
65                 if (in2 >> b) hasB = true;
66                 else hasB = false;
67             }
68         }
69
70         while (countA < blockSize && hasA)
71         {
72             out << a << " ";
73             countA++;
74             if (in1 >> a) hasA = true;
75             else hasA = false;
76         }
77
78         while (countB < blockSize && hasB)
79         {
80             out << b << " ";
81             countB++;
82             if (in2 >> b) hasB = true;
83             else hasB = false;
84         }
85     }
```

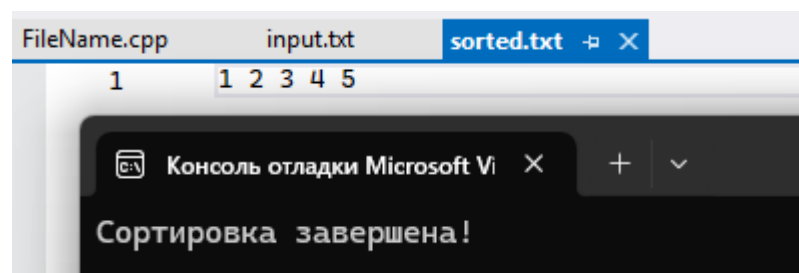
```

85     }
86
87     in1.close();
88     in2.close();
89     out.close();
90 }
91
92 // Подсчёт количества чисел в файле
93 int countNumbers(const char* filename)
94 {
95     ifstream in(filename);
96     int count = 0, x;
97     while (in >> x) {
98         count++;
99     }
100     in.close();
101     return count;
102 }
103
104 int main() {
105     setlocale(LC_ALL, "ru");
106     const char* input = "input.txt";
107     const char* output = "sorted.txt";
108     const char* temp1 = "temp1.txt";
109     const char* temp2 = "temp2.txt";
110
111     int n = countNumbers(input);
112     int blockSize = 1;
113
114     while (blockSize < n) {
115         split(input, temp1, temp2, blockSize);
116         merge(temp1, temp2, input, blockSize);
117         blockSize *= 2;
118     }
119
120     rename(input, output);
121
122     cout << "Сортировка завершена!" << endl;
123     return 0;
124 }

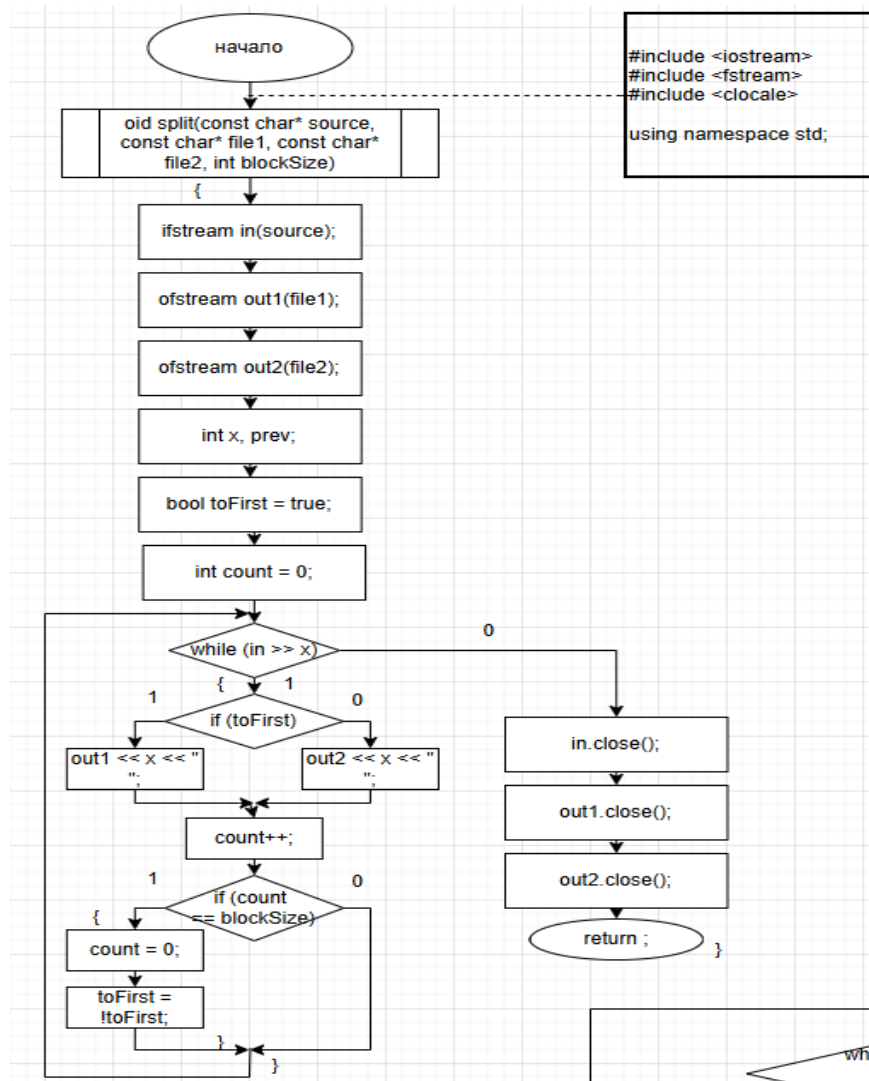
```

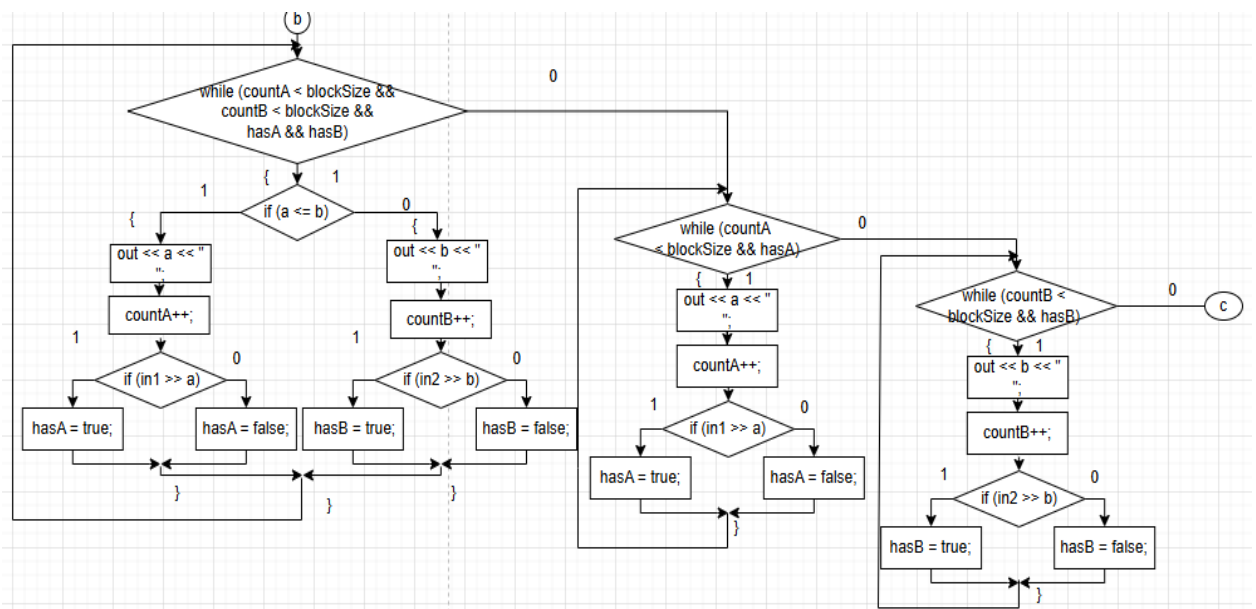
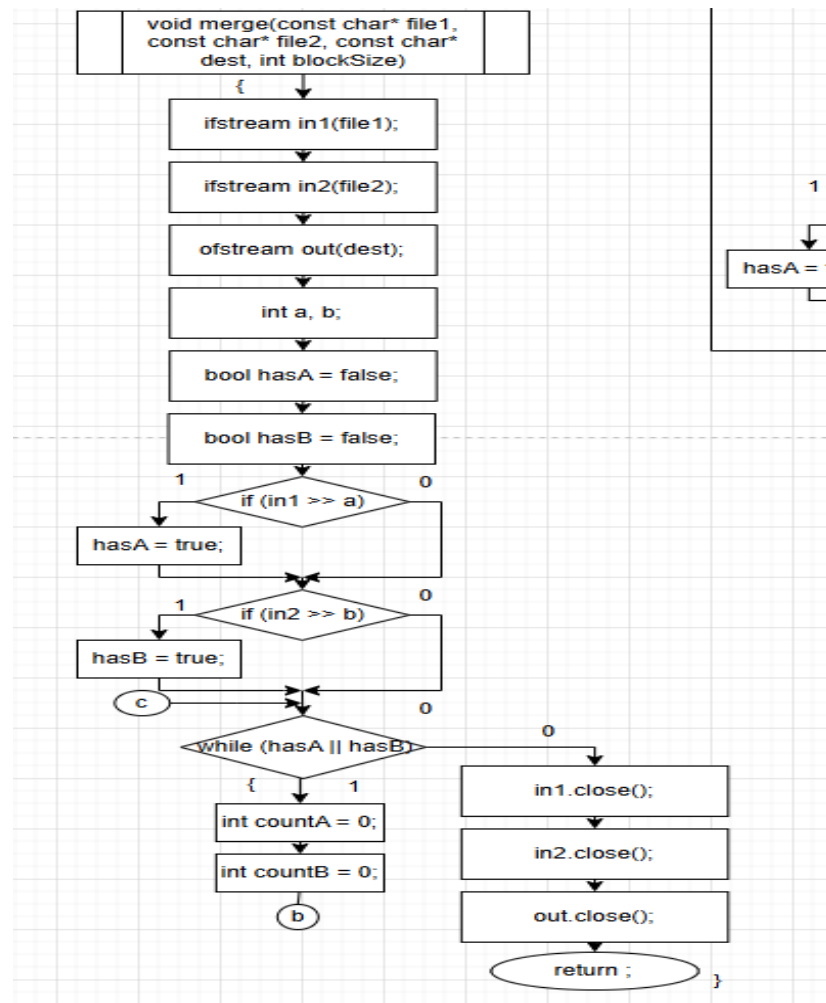
4.2 Результат работы программы

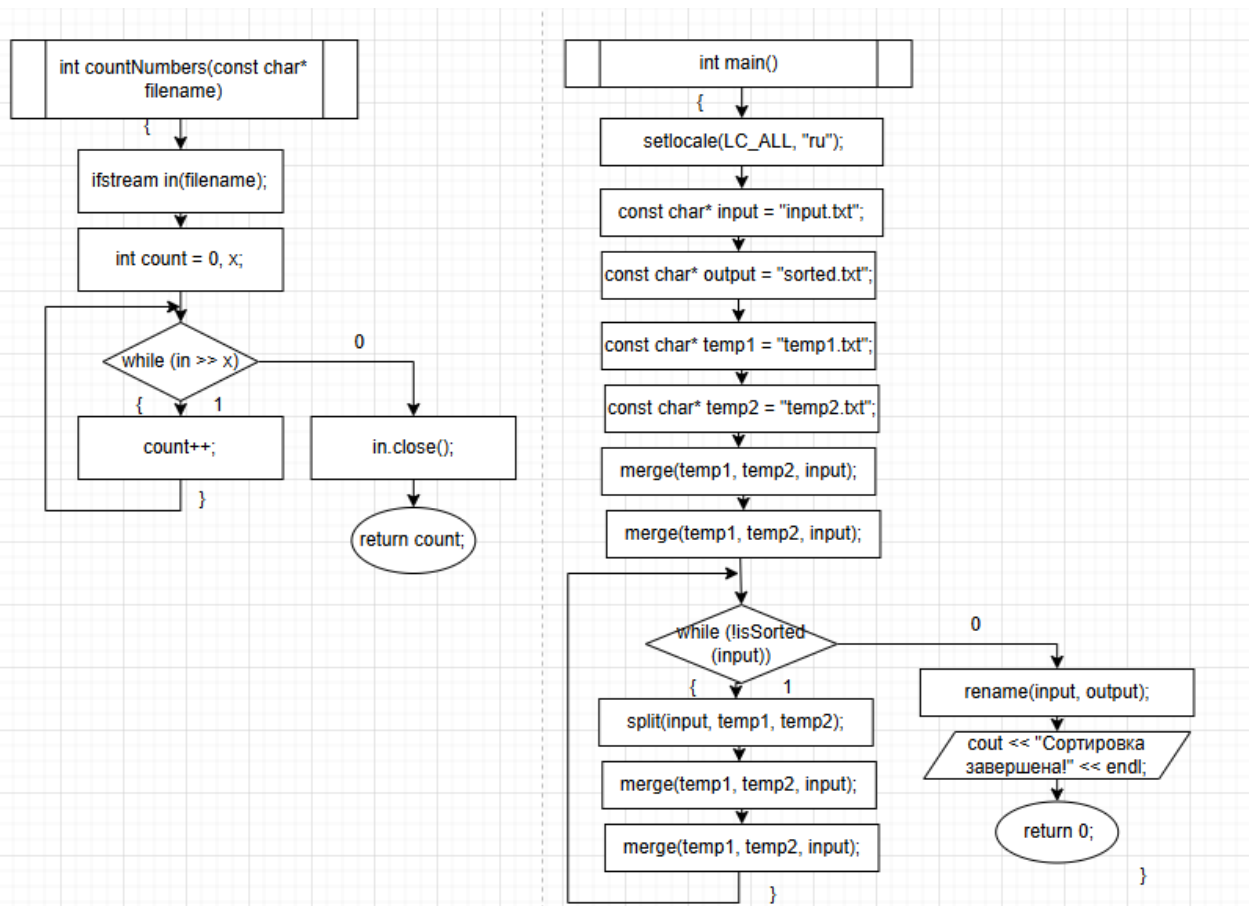
Исходный массив: 5 2 1 3 4



4.3 Блок-схема







5 Ссылка на github

ссылка на github - <https://github.com/MAKSPOWERO/mas1>