



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего  
образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

---

**ОТЧЕТ  
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1**

Сортировка числового файла с помощью битового массива  
по дисциплине  
«Структуры и алгоритмы обработки данных»

Выполнила студентка группы ИКБО-01-21

Цаплина Е.А.

Принял преподаватель

Сартаков М.В.

Практическая  
работа выполнена

«\_\_»\_\_\_\_\_2022 г.

\_\_\_\_\_

«Зачтено»

«\_\_»\_\_\_\_\_2022 г.

\_\_\_\_\_

Москва 2022

## СОДЕРЖАНИЕ

СОДЕРЖАНИЕ .....	2
1 ЦЕЛЬ РАБОТЫ .....	3
2 ХОД РАБОТЫ .....	4
2.1 Задание 1.а.....	4
2.2 Задание 1.б .....	5
2.3 Задание 1.в.....	6
2.4 Задание 2.а.....	7
2.5 Задание 2.б .....	8
2.6 Задание 2.в.....	9
2.7 Задание 3.а и 3.б.....	10
3 ВЫВОД .....	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	14

## **1 ЦЕЛЬ РАБОТЫ**

Освоить приёмы работы с битовым представлением беззнаковых целых чисел, реализовать эффективный алгоритм внешней сортировки на основе битового массива.

## 2 ХОД РАБОТЫ

### 2.1 Задание 1.а

#### Формулировка задачи:

Установить 5-й бит произвольного целого числа в 0. Проверить работоспособность программы.

#### Математическая модель решения:

Для того, чтобы установить 5-й бит числа в 0, воспользуемся 8-ми разрядной маской, изначально равной 1 (00000001). Далее с помощью побитового сдвига, перемещаем единицу в маске на 5-ю позицию, после чего маску необходимо инвертировать и провести операцию поразрядной конъюнкции с исходным двоичным числом. Тогда все биты, кроме того, куда мы изначально поместили 1 сохраняют свое значения, а необходимый бит станет 0.

#### Код программы:

Листинг 1.1 – Код программы

```
// пункт а
void first_foo() {
    unsigned char x=255; // 8-разрядное двоичное число 11111111
    unsigned char maska = 1; // 1=00000001 - 8-разрядная маска
    x = x & (~ (maska<<4)); // результат x=239
    cout << (int) x << endl;
}
```

#### Результаты тестирования:

Результаты тестирования представлены на Рис.1 – Рис.2.

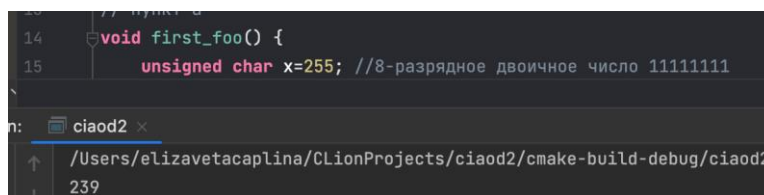


Рисунок 1 - Результат тестирования на числе 255

```
14 void first_foo() {
15     unsigned char x=16; //8-разрядное двоичное число 00010000
    first_foo
    ciaod2 x
    /Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/ciaod2
    0
    Process finished with exit code 0
```

Рисунок 2 - Результат тестирования на числе 16

## 2.2 Задание 1.6

### Формулировка задачи:

Установить 7-й бит произвольного целого числа в 1. Проверить работоспособность программы.

### Математическая модель решения:

Данная задача решается аналогично 1. А, но без инвертирования маски, кроме того, вместо поразрядной конъюнкции производим дизъюнкцию.

Листинг 1.2 – Код программы

```
// пункт 6
void second_foo() {
    unsigned char x=0; //8-разрядное двоичное число 00000000
    unsigned char maska = 1; //1=00000001 - 8-разрядная маска
    x = x | (maska<<6); //результат x=64
    cout << (int) x << endl;
}
```

### Результаты тестирования:

Результаты тестирования показаны на Рис.3 – Рис. 4.

```
6 // пункт 6
7 void second_foo() {
8     unsigned char x=0; //8-разрядное двоичное число 00000000
    main
    ciaod2 x
    /Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/cia
    64
```

Рисунок 3 - Результат тестирования на числе 1

```
7 void second_foo() {
8     unsigned char x=191; //8-разрядное двоичное число 10111111
    second_foo
}

Run: ciaod2 x
/Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/ciaod2
255
Process finished with exit code 0
```

Рисунок 4 - Результат тестирования на числе 191

## 2.3 Задание 1.в

### Формулировка задачи:

Реализовать приведённый в задании код.

### Математическая модель решения:

Выводим двоичное представление числа с помощью двоичной маски путем ее битового сдвига.

*Листинг 1.3 – Код программы*

```
// пункт в
void third_foo(){
    unsigned int x = 25;
    const int n = sizeof(int)*8; // =32 - кол-во разрядов в инте
    unsigned maska = (1 << n - 1); // 1 в 32 разряде
    cout << "Начальный вид маски: " << bitset<n> (maska) << endl;
    cout << "Результат: ";
    for (int i = 1; i <= n; ++i) {
        cout << ((x & maska) >> (n - i));
        maska = maska >> 1;
    }
    cout << endl;
}
```

### Результаты тестирования:

Результаты тестирования представлены на Рис.5 – Рис.6.

```
20 // пункт в
21 void third_foo(){
22     unsigned int x = 25;
    third_foo
}

Run: ciaod2 x
/Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/ciaod2
Начальный вид маски: 10000000000000000000000000000000
Результат: 0000000000000000000000000000000011001
```

Рисунок 5 - Результат тестирования на числе 25

```
21 void third_foo(){
22     unsigned int x = 255;
    third_foo
n: ciaod2 x
/Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/ciaod2
Начальный вид маски: 10000000000000000000000000000000
Результат: 0000000000000000000000000000000011111111
Process finished with exit code 0
```

Рисунок 6 - Результат тестирования на числе 255

## 2.4 Задание 2.а

### Формулировка задачи:

Отсортировать набор из не более чем 8-ми уникальных цифр со значениями от 0 до 7 битовым массивом в виде числа типа unsigned char.

### Математическая модель решения:

Получаем размер массива, затем в цикле во временную переменную получаем определенное число, далее с помощью битового сдвига помещаем единицу на позицию равную введенному числу. После окончания ввода проходим циклом по битовому массиву начиная с 0 разряда и выводим номера разрядов, на которых стоит единица.

#### Листинг 1.4 – Код программы

```
// пункт а (сортировка с помощью битового массива)
void first_foo() {
    int n;
    int buff;
    unsigned char x = 0;

    cout << "Введите кол-вл чисел (<=8): ";
    cin >> n;
    cout << "Введите числа (от 0 до 7): ";
    for (int i = 0; i < n; ++i) {
        cin >> buff;
        x = x | (1<<(buff));
    }
    bitset<8> b = bitset<8>(x);
    cout << "Отсортированный массив: " << endl;
    for (int i = 0; i < 8; ++i) {
        if (b[i] == 1) cout << i << " ";
    }
}
```

### Результаты тестирования:

Результаты тестирования показаны на Рис.7 – Рис.8.

```
/Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/ciaod2
Введите кол-вл чисел (<=8): 5
Введите числа (от 0 до 7): 2 5 1 6 3
Отсортированный массив:
1 2 3 5 6
Process finished with exit code 0
```

Рисунок 7 - Результат тестирования программы

```
/Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/ciaod2
Введите кол-вл чисел (<=8): 8
Введите числа (от 0 до 7): 2 3 6 4 1 7 0 5
Отсортированный массив:
0 1 2 3 4 5 6 7
Process finished with exit code 0
```

Рисунок 8 - Результат тестирования программы

## 2.5 Задание 2.6

### Формулировка задачи:

Адаптировать код предыдущей задачи для набора из 64-х чисел (со значениями от 0 до 63) с битовым массивом в виде числа типа unsigned long long.

### Математическая модель решения:

Ход решения аналогичен коду предыдущей задачи, за исключением типа данных для хранения числа (теперь unsigned long long).

#### Листинг 1.5 – Код программы

```
// пункт б (для 64 бит)
void second_foo() {
    int n;
    int buff;
    unsigned long long x = 0;

    cout << "Введите кол-вл чисел (<=64): ";
    cin >> n;
    cout << "Введите числа (от 0 до 63): ";
    for (int i = 0; i < n; ++i) {
        cin >> buff;
        x = x | ((unsigned long long) 1 << (buff));
    }
    bitset<64> b = bitset<64>(x);
    cout << "Отсортированный массив: " << endl;
    for (int i = 0; i < 64; ++i) {
        if (b[i] == 1) cout << i << " ";
    }
}
```



```
}  
}
```

### Результаты тестирования:

Результаты тестирования показаны на Рис.9.

```
/Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/ciaod2  
Введите кол-во чисел (<=64): 20  
Введите числа (от 0 до 63): 23 12 0 30 55 1 2 8 4 7 28 27 10 17 39 60 58 54 53 51  
Отсортированный массив:  
0 1 2 4 7 8 10 12 17 23 27 28 30 39 51 53 54 55 58 60
```

Рисунок 9 - Результат тестирования программы

## 2.6 Задание 2.в

### Формулировка задачи:

Исправьте программу задания 2.б, чтобы для сортировки набора из 64-х чисел использовалось не одно число типа unsigned long long, а линейный массив чисел типа unsigned char.

### Математическая модель решения:

Ход решения аналогичен коду предыдущей задачи, однако теперь число хранится в массиве типа unsigned char

*Листинг 1.6 – Код программы*

```
void third_foo() {  
    int n;  
    int buff;  
    unsigned char x[8];  
  
    cout << "Введите кол-во чисел (<=64): ";  
    cin >> n;  
  
    for (int i = 0; i < 8; ++i) {  
        x[i] = 0;  
    }  
  
    cout << "Введите числа (от 0 до 63): ";  
    for (int i = 0; i < n; ++i) {  
        cin >> buff;  
        x[buff / 8] = x[buff / 8] | (1 << (buff % 8));  
    }  
  
    for (int i = 0; i < 8; i++)  
        for (int j = 0; j < 8; j++)  
            if ((1 << j) & x[i]) cout << i * 8 + j << " ";  
}
```

## Результаты тестирования:

Результаты тестирования показаны на Рис.10.

```
/Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/ciaod2
Введите кол-вл чисел (<=64): 20
Введите числа (от 0 до 63): 23 12 0 30 55 1 2 8 4 7 28 27 10 17 39 60 58 54 53 51
Отсортированный массив:
0 1 2 4 7 8 10 12 17 23 27 28 30 39 51 53 54 55 58 60
```

Рисунок 10 - Результаты работы программы

## 2.7 Задание 3.а и 3.б

### Формулировка задачи:

Реализуйте задачу сортировки числового файла с заданными условиями. Добавьте в код возможность определения времени работы программы. Определите программно объём оперативной памяти, занимаемый битовым массивом.

### Математическая модель решения:

Сначала необходимо получить количество чисел с клавиатуры, далее сгенерировать массив, заполнить его случайными различными числами и записать элементы массива в файл. После чего производится описанная выше сортировка битовым массивом (вектором типа `unsigned char`) и снова запись в файл. Кроме того считается время работы программы с помощью функции `clock()` и затраченная память.

Листинг 1.7 – Код программы

```
#include <iostream>
#include <random>
#include <fstream>
#include <ctime>

using namespace std;
void generate_file() {
    int n;
    ofstream out;

    auto rng = default_random_engine {}; // генерация

    cout << "Введите количество чисел: ";
    cin >> n;

    vector<int> nums = vector<int>(n);

    out.open("input.txt");

    iota(nums.begin(), nums.end(), 0); // заполнили массив 0
    shuffle(nums.begin(), nums.end(), rng); // перетасуем массив
```

```

        // запись в файл
        for(int i = 0; i < n; i++) {
            out << nums[i] << "\n";
        }
        out.close();
    }

int main() {
    // генерируем файл
    generate_file();

    // засекаем время
    unsigned int start_time = clock();

    ifstream in;
    in.open("input.txt");

    const int kshift = 7;
    unsigned char block = 0;
    int num = 0;
    vector<unsigned char> array = vector<unsigned char>(0);

    while(in >> num) {
        int block_n = num / 8;
        int sz = array.size() - 1;
        if (block_n > sz) {
            for(int j = 0; j < block_n - sz; j++) array.push_back(block);
        }
        array[block_n] = array[block_n] | 1 << (kshift - num % 8);
    }
    in.close();

    ofstream out;
    out.open("output.txt");

    int len = array.size() * 8;
    for(int i = 0; i < len; i++) {
        if ((array[i / 8] >> (kshift - i % 8)) & 1) out << i << "\n";
    }
    out.close();

    unsigned int end_time = clock();
    unsigned int search_time = end_time - start_time;
    int kbits = (array.size() * sizeof(unsigned char)) / 1024;

    cout << "Потраченное время: " << search_time / 1000 << "ms" << "\n";
    cout << "Потраченная оперативная память: " << kbits << " kb" << "\n";
}

```

## Результаты тестирования:

Результаты тестирования показаны на Рис.11-Рис.12.

```
un: ciaod2 x
/Users/elizavetacaplina/CLionProjects/ciaod2/cmake-build-debug/ciaod2
Введите количество чисел: 10000000
Потраченное время: 3091ms
Потраченная оперативная память: 1220 kb
Process finished with exit code 0
```

Рисунок 11 - Результат работы программы

The file size (78,89 MB) ...			0
1	48270	✓	1
2	2118013		2
3	4702500		3
4	2845313		4
5	7752998		5
6	8814767		6
7	8375304		7
8	3508783		8
9	488171		9
10	8967529		10
11	2276749		11
12	9795169		12
13	6650308		13
14	1279072		14
15	1396807		15
16	1276833		16
17	1250019		17
18	8094379		18
19	1550568		19
20	3524886		20
Text	Large File Editor		21

Рисунок 12 - Результат работы программы

### **3      ВЫВОД**

В процессе выполнения практической работы мной была освоена тема побитовых сдвигов, так же я впервые познакомилась со способом сортировки чисел битовым массивом. Мной были освоены приёмы работы с битовым представлением беззнаковых целых чисел, реализован эффективный алгоритм внешней сортировки на основе битового массива.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Страуструп Б. Программирование. Принципы и практика с использованием C++. 2-е изд., 2016.
2. Документация по языку C++ [Электронный ресурс]. URL: <https://docs.microsoft.com/ruru/cpp/cpp/> (дата обращения 01.09.2021).
3. Курс: Структуры и алгоритмы обработки данных. Часть 2 [Электронный ресурс]. URL: <https://online-edu.mirea.ru/course/view.php?id=4020> (дата обращения 01.09.2021).