



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

---

Институт информационных технологий (ИТ)

Кафедра математического обеспечения и стандартизации информационных технологий  
(МОСИТ)

## **ОТЧЁТ ПО ПРАКТИЧЕСКИМ РАБОТАМ**

**по дисциплине «Структуры и алгоритмы обработки данных (часть  
2/2)»**

Выполнил студент группы ИКБО-01-22

Аблаева А.Т.

Принял  
*Ассистент*

Ермаков С.Р.

Практические работы выполнены

«\_\_»\_\_\_\_\_2023 г.

(подпись студента)

«Зачтено»

«\_\_»\_\_\_\_\_2023 г.

(подпись преподавателя)

Москва 202

## СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ .....	3
2 ЗАДАНИЕ 1 .....	3
2.1 Формулировка задачи.....	3
2.2 Математическая модель решения .....	3
2.3 Код программы .....	4
2.4 Результаты тестирования .....	4
3 ЗАДАНИЕ 2 .....	5
3.1 Формулировка задачи.....	5
3.2 Математическая модель решения .....	6
3.3 Код программы .....	7
3.4 Результаты тестирования .....	9
4 ЗАДАНИЕ 3 .....	10
4.1 Формулировка задачи.....	10
4.2 Математическая модель решения .....	11
4.3 Код программы .....	11
4.4 Результат тестирования.....	12
5 ВЫВОД .....	12
6 СПИСОК ИСТОЧНИКОВ.....	12

## 1 ЦЕЛЬ РАБОТЫ

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

## 2 ЗАДАНИЕ 1

### 2.1 Формулировка задачи

Задание 1:

**1.а.** Реализуйте вышеприведённый пример, проверьте правильность результата в том числе и на других значениях  $x$ .

**1.б.** Реализуйте по аналогии с предыдущим примером установку 7-го бита числа в единицу.

**1.в.** Реализуйте код листинга 1, объясните выводимый программой результат.

### 2.2 Математическая модель решения

В задании 1.а необходимо установить 5-й бит числа в 0. Алгоритм: побитово перемножаем заданное число с «маской», в которой все разряды, кроме 5, равны 1.

В задании 1.б необходимо установить 7-й бит числа в 1. Алгоритм: побитово складываем заданное число с «маской», в которой все разряды, кроме 7, равны 0.

В задании 1.в представлен код, необходимо объяснить результат его выполнения. Задается целочисленное число  $x$  и число  $n$ , хранящее количество битов числа  $x$ . Также задается число  $mask$ , в котором  $n$  разрядов и в старшем бит установлен в 1. Выводится битовое представление числа  $mask$ . Далее в цикле для каждого из  $n$  разрядов получаем побитовое умножение  $x$  и  $mask$ , со сдвигом до младшего разряда. Выводим получившееся число, представляющее собой бит числа  $x$ . В  $mask$  сдвигаем 1 на 1 разряд в сторону младшего разряда. В результате работы программы получим битовое представление числа  $x$ .

## 2.3 Код программы

```
int f_a(unsigned char x) { //установка 5-го бита числа в 0 – уменьшенное на 16 значение числа или то же самое число
    unsigned char maska = 1;
    x = x & ~(maska << 4);
    return x;
}
```

Рис. 1 – код задания 1.а

```
int f_b(unsigned char x) { //установка 7-го бита числа в 1 – увеличенное на 64 значение числа или то же самое число
    unsigned char maska = 1;
    x = x | (maska << 6);
    return x;
}
```

Рис. 2 – код задания 1.б

```
void f_c() { //реализация кода листинга
    unsigned int x = 25;
    const int n = sizeof(x) * 8; //количество битов числа – 32
    unsigned maska = (1 << (n - 1)); //установка 1 в старший разряд разрядного числа
    cout << "Начальный вид маски: " << bitset<n>(maska) << endl; //создание объекта класса bitset – инициализация битов соответствующего числа
    cout << "Результат: ";
    for (int i = 1; i <= n; i++) {
        cout << ((x & maska) >> (n - i)); //вывод битов переменной x слева направо
        maska = maska >> 1; //сдвиг 1 в маске влево на бит
    }
    cout << endl;
}
```

Рис. 3 – код задания 1.в

## 2.4 Результаты тестирования

```
Выберите задание: 1 2 3
Для выхода из меню введите 0
1
Выберите подзадание: a b c
a
Введите неотрицательное целое число до 255:
54
Число с 5-м нулевым битом:
38
```

Рис. 4 – результат работы задания 1.а с параметром 54

```
Выберите задание: 1 2 3
Для выхода из меню введите 0
1
Выберите подзадание: a b c
b
Введите неотрицательное целое число до 255:
32
Число с 7-м единичным битом:
96
```

Рис. 5 – результат работы задания 1.б с параметром 32

```
Выберите задание: 1 2 3
Для выхода из меню введите 0
1
Выберите подзадание: a b c
c
Начальный вид маски: 10000000000000000000000000000000
Результат: 0000000000000000000000000000000011001
```

Рис. 6 – результат работы задания 1.в

## 3 ЗАДАНИЕ 2

### 3.1 Формулировка задачи

Пусть даны не более 8 чисел со значениями от 0 до 7, например, {1, 0, 5, 7, 2, 4}.

Подобный набор чисел удобно отразить в виде 8-разрядной битовой последовательности **11101101**. В ней единичные биты показывают *наличие* в исходном наборе числа, равного номеру этого бита в последовательности (нумерация с 0 слева). Т.о. индексы единичных битов в битовом массиве – это и есть числа исходной последовательности.

Последовательное считывание бит этой последовательности и вывод индексов единичных битов позволит естественным образом получить исходный набор чисел *в отсортированном виде* – {0, 1, 2, 4, 5, 7}.

Задание 2:

**2.а.** Реализуйте вышеописанный пример с вводом произвольного набора до 8-ми чисел (со значениями от 0 до 7) и его сортировкой битовым массивом в виде числа типа unsigned char. Проверьте работу программы.

**2.б.** Адаптируйте вышеприведённый пример для набора из 64-х чисел (со значениями от 0 до 63) с битовым массивом в виде числа типа unsigned long long.

**2.в.** 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.

### 3.2 Математическая модель решения

В задании 2.а необходимо отсортировать массив (максимум 8 чисел со значениями от 0 до 7) используя битовую сортировку. С помощью маски, изначально равной 1, сдвига маски влево на необходимое количество бит и побитового сложения с 0, записываем в соответствующий бит числа типа `unsigned char` единицу, чтобы записать введенное число. Далее в маске устанавливаем 1 в старший разряд. При прохождении по разрядам числа слева направо и побитовом умножении числа с маской, которая сдвигается вправо на разряд при каждой итерации, выводим число, соответствующее номеру разряда слева направо, начиная с 0. Таким образом получаем отсортированный массив чисел.

В задании 2.б необходимо отсортировать массив (максимум 64 числа со значениями от 0 до 63) используя битовую сортировку. С помощью маски, изначально равной 1, сдвига маски влево на необходимое количество бит и побитового сложения с 0, записываем в соответствующий бит числа типа `unsigned long long` единицу, чтобы записать введенное число. Далее в маске устанавливаем 1 в старший разряд. При прохождении по разрядам числа слева направо и побитовом умножении числа с маской, которая сдвигается вправо на разряд при каждой итерации, выводим число, соответствующее номеру разряда слева направо, начиная с 0. Таким образом получаем отсортированный массив чисел.

В задании 2.в необходимо отсортировать массив используя битовую сортировку. Составляем массив, состоящий из чисел типа `unsigned char`. При вводе числа находим индекс элемента массива, в который нужно его записать, с помощью нахождения целой части при делении на 8, и номер бита, который установится в 1, с помощью нахождения остатка от деления на 8. Проходимся циклом по элементам массива. Во вложенном цикле проходимся по разрядам числа слева направо и побитово умножаем числа с маской, которая сдвигается вправо на разряд при каждой итерации. Если получаем 1, то выводим число,

соответствующее номеру разряда слева направо, начиная с 0, сложенному с произведением 8 и индекса элемента массива. Таким образом получаем отсортированный массив чисел.

### 3.3 Код программы

```
void s_a() {
    unsigned char a = 0;
    unsigned char maska = 1;
    float x;
    int i = 0;
    cout << "Введите до 8 целых чисел от 0 до 7: " << endl;
    cout << "Чтобы прекратить ввод, введите -1" << endl;
    while (i < 8) {
        cin >> x;
        if (x == -1)
            break;
        while (x > 8 || x < 0 || (x - (int)x) > 0) {
            cout << "Введите до 8 целых чисел от 0 до 7: " << endl;
            cout << "Чтобы прекратить ввод, введите -1" << endl;
            cin >> x;
        }
        a = a | (maska << (8 - (int)x - 1));
        maska = 1;
        i++;
    }
    maska = maska << 7;
    cout << "Отсортированная последовательность: " << endl;
    for (int i = 1; i <= 8; i++) {
        if (((a & maska) >> (8 - i)) == 1) {
            cout << i - 1 << " ";
        }
        maska = maska >> 1;
    }
    cout << endl << "Битовый массив: " << bitset<8>(a) << endl;
}
```

Рис. 7 – код задания 2.а

```

void s_b() {
    unsigned long long a = 0;
    unsigned long long maska = 1;
    float x;
    int n;
    srand(time(NULL));
    cout << "Введите до 64 целых чисел от 0 до 63: " << endl;
    cout << "Ввести числа вручную - 1, случайные числа - 2" << endl;
    cout << "Чтобы прекратить ввод, введите -1" << endl;
    cin >> n;
    for (int i = 0; i < 64; i++) {
        switch (n) {
            case 1:
                cin >> x;
                if (x == -1)
                    break;
                while (x < 0 || x > 63 || (x - (int)x) > 0) {
                    cout << "Введите до 64 целых чисел от 0 до 63: " << endl;
                    cout << "Чтобы прекратить ввод, введите -1" << endl;
                    cin >> x;
                }
                break;
            case 2:
                x = rand() % 64;
                break;
        }
        if (x == -1)
            break;
        a = a | (maska << (64 - (int)x - 1));
        maska = 1;
    }
}

```

Рис. 8 – код задания 2.6 (1/2)

```

maska = maska << 63;
cout << "Отсортированная последовательность: ";
for (int i = 1; i <= 64; i++) {
    if (((a & maska) >> (64 - i)) == 1) {
        cout << i - 1 << " ";
    }
    maska = maska >> 1;
}
cout << endl << "Битовый массив: " << bitset<64>(a) << endl;
}

```

Рис. 9 – код задания 2.6 (2/2)



```

void s_c() {
    vector<unsigned char> mas = { 0 };
    int x;
    cout << "Введите целые числа: " << endl;
    cout << "Чтобы прекратить ввод, введите -1" << endl;
    cin >> x;
    while (x != -1) {
        while (mas.size() - 1 < (x / 8)) {
            mas.push_back(0);
        }
        mas[x / 8] = mas[x / 8] | (1 << (8 - (x % 8) - 1));
        cin >> x;
    }
    cout << "Отсортированный массив чисел: " << endl;
    for (int i = 0; i < mas.size(); i++) {
        for (int j = 0; j <= 7; j++) {
            if (((mas[i] & (1 << (8 - j - 1))) >> (8 - j - 1)) == 1) {
                cout << i * 8 + j << " ";
            }
        }
    }
    cout << endl;
}

```

Рис. 10 – код задания 2.в

### 3.4 Результаты тестирования

```

Выберите задание: 1 2 3
Для выхода из меню введите 0
2
Выберите подзадание: a b c
a
Введите до 8 целых чисел от 0 до 7:
Чтобы прекратить ввод, введите -1
1 2 3 6 2 5 8 -1
Отсортированная последовательность:
1 2 3 5 6
Битовый массив: 01110110

```

Рис. 11 – результат работы задания 2.а

```

Выберите задание: 1 2 3
Для выхода из меню введите 0
2
Выберите подзадание: a b c
b
Введите до 64 целых чисел от 0 до 63:
Ввести числа вручную - 1, случайные числа - 2
Чтобы прекратить ввод, введите -1
1
12 32 45 0 56 61 12 3 25 7 8 -1
Отсортированная последовательность: 0 3 7 8 12 25 32 45 56 61
Битовый массив: 100100011000100000000000001000000100000000000010000000000100000100

```

Рис. 12 – результат работы задания 2.б

```

Выберите задание: 1 2 3
Для выхода из меню введите 0
2
Выберите подзадание: a b c
c
Введите целые числа:
Чтобы прекратить ввод, введите -1
123 473724 12313 101 2 321 47 -1
Отсортированный массив чисел:
2 47 101 123 321 12313 473724

```

Рис. 13 – результат работы задания 2.в

## 4 ЗАДАНИЕ 3

### 4.1 Формулировка задачи

Реализовать высокоэффективную сортировку большого объёма числовых данных в файле можно на идее битового массива. Достаточно один раз считать содержимое файла, заполнив при этом в памяти ЭВМ битовый массив и на его основе быстро сформировать содержимое выходного файла в уже отсортированном виде.

При использовании битового массива для представления сортируемых чисел, программу можно представить как последовательность из трех подзадач:

- а) Создание битового массива с нулевыми исходными значениями.
- б) Считывание целых чисел из файла и установка в 1 соответствующих бит массива.
- в) Формирование упорядоченного выходного файла путём последовательной проверки бит массива и вывода в файл номеров (индексов) тех бит, которые установлены в 1.

Задание 3:

**3.а.** Реализуйте задачу сортировки числового файла с заданными условиями. Добавьте в код возможность определения времени работы программы.

**3.б.** Определите программно объём оперативной памяти, занимаемый битовым массивом.

## 4.2 Математическая модель решения

В задании 3.а необходимо считать из файла  $10^7$  чисел, отсортировать их и занести в файл. Составляем массив, состоящий из чисел типа unsigned char, заполненный нулями. При считывании числа из файла находим индекс элемента массива, в который нужно его записать, с помощью нахождения целой части при делении на 8, и номер бита, который установится в 1, с помощью нахождения остатка от деления на 8. Проходимся циклом по элементам массива. Во вложенном цикле проходимся по разрядам числа слева направо и побитово умножаем числа с маской, которая сдвигается вправо на разряд при каждой итерации. Если получаем 1, то записываем в файл число, соответствующее номеру разряда слева направо, начиная с 0, сложенному с произведением 8 и индекса элемента массива. Таким образом получаем отсортированный массив чисел.

В задании 3.б необходимо определить программный объем оперативной памяти, занимаемый битовым массивом. Это можно сделать с помощью метода sizeof().

## 4.3 Код программы

```
void t() {
    vector<unsigned char> mas(1250000);
    int x;
    clock_t start = clock();
    ifstream fin("nums.txt");
    ofstream fout("num_sort.txt");
    while (!fin.eof()) {
        fin >> x;
        x--;
        mas[x / 8] = mas[x / 8] | (1 << (8 - (x % 8) - 1));
    }
    for (int i = 0; i < mas.size(); i++) {
        for (int j = 0; j <= 7; j++) {
            if (((mas[i] & (1 << (8 - j - 1))) >> (8 - j - 1)) == 1) {
                fout << i * 8 + j + 1 << " ";
            }
        }
    }
    fin.close();
    fout.close();
    clock_t end = clock();
    double seconds = (double)(end - start) / CLK_TCK;
    cout << "Время работы программы: " << seconds << " секунд" << endl;
    cout << "Объем оперативной памяти, занимаемой битовым массивом: " << sizeof(mas) / (1024.0 * 1024.0) << " MB" << endl;
}
```

Рис. 14 – код задания 3

#### 4.4 Результат тестирования

```
Выберите задание: 1 2 3
Для выхода из меню введите 0
3
Время работы программы: 146.358 секунд
Объем оперативной памяти, занимаемой битовым массивом: 3.05176e-05 MB
Выберите задание: 1 2 3
```

Рис. 15 – результат работы задания 3

### 5 ВЫВОД

В первой практической работе я освоила приёмы работы с битовым представлением беззнаковых целых чисел и реализовала эффективный алгоритм внешней сортировки на основе битового массива.

### 6 СПИСОК 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/ru-ru/cpp/cpp/> (дата обращения 20.09.2023).
3. Курс: Структуры и алгоритмы обработки данных. Часть 2 [Электронный ресурс]. URL: <https://online-edu.mirea.ru/course/view.php?id=4020> (дата обращения 22.09.2023).