



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

---

Институт информационных технологий (ИИТ)  
Кафедра МОиСИТ

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5.1**  
**«Битовые операции. Сортировка числового файла с помощью**  
**битового массива»**  
**по дисциплине**  
**«Структуры и алгоритмы обработки данных»**

Выполнил студент группы ИКБО-10-24

Бикташев И. И.

Практическую работу выполнил «\_\_»\_\_\_\_\_2025 г.

«Зачтено» «\_\_»\_\_\_\_\_2025 г.

Москва 2025

## Задание 1

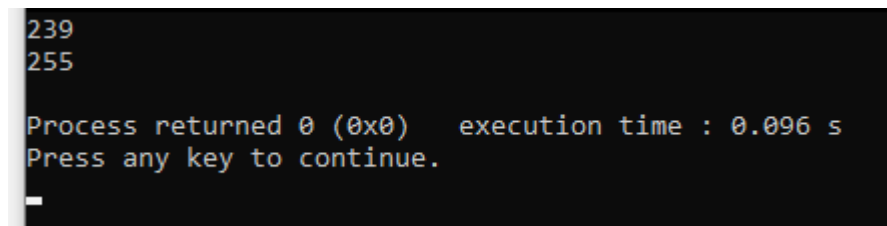
Программа, представленная на листинге 1, берёт число 255 в двоичном виде (11111111), заменяет ему 5 бит на ноль и выводит на экран. После программа заменяют числу 191 (10111111 в двоичном виде) 7 бит на единицу и выводит результат на экран.

Листинг 1

```
#include <iostream>

using namespace std;

int main()
{
    unsigned char x = 0b11111111; //8-разрядное двоичное число 11111111 (255)
    unsigned char mask = 1; //1=00000001 - 8-разрядная маска
    x &= ~(mask << 4); // 5 бит в 0
    cout << (int)x << endl;
    x = 0b10111111;
    x |= mask << 6; // 7 бит в 1
    cout << (int)x << endl;
    return 0;
}
```



```
239
255

Process returned 0 (0x0)   execution time : 0.096 s
Press any key to continue.
_
```

Рисунок 1 - Вывод программы

Реализовал код представленный на листинге 2

Листинг 2

```
#include <cstdlib>
#include <iostream>
#include <Windows.h>
#include <bitset>
using namespace std;

int main()
{
    SetConsoleCP (1251);
    SetConsoleOutputCP (1251);

    unsigned int x = 25;
    const int n = sizeof(int)*8; //=32 количество разрядов в числе типа int
    unsigned maska = (1 << n - 1); //1 в старшем бите 32-разрядной сетки

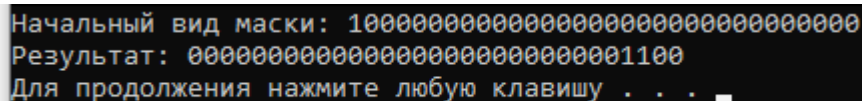
    cout << "Начальный вид маски: " << bitset<n>(maska) << endl;
```

```

cout << "Результат: ";

for (int i = 1; i < n; i++) //32 раза
{
    cout << ((x & maska) >> (n - i));
    maska = maska >> 1; // смещение 1 в маске на разряд вправо
}
cout << endl;
system("pause");
return 0;
}

```



```

Начальный вид маски: 10000000000000000000000000000000
Результат: 000000000000000000000000000000001100
Для продолжения нажмите любую клавишу . . .

```

Рисунок 2 - Вывод программы

Данная программа выводит двоичное представление числа 25 в виде 32 разрядного числа на экран с помощью цикла. Каждый проход цикла выполняется операция или с маской (у которой только 1 в старшем бите 32-разрядной сетки) сдвинутой вправо на количестве итераций цикла и числом 25 и после выводит результат, смещенный на  $n - i$  вправо.

## Задание 2

Сортировка с помощью массива битов реализуется следующим образом: перебираются все элементы исходного массива и в битовом массиве, представленном в виде числа типа `unsigned char`, бит с индексом равным данному элементу (биты нумеруются в данном случае слева направо) меняется на единицу.

Листинг 3

```

#include <iostream>

using namespace std;

int main()
{
    const int n = 8;
    int numbers[] {1, 6, 5, 7, 2, 3};
    unsigned char sorted = 0;

    for (auto i : numbers) {
        sorted |= 1 << (n - i - 1);
    }

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

```

```

        if ((sorted & (mask << (n - i - 1))) > 0) {
            cout << i << ", ";
        }
    }
    return 0;
}

```

В листинге 4 представлен вариант программы выше, поддерживающей сортировку чисел в диапазоне от 0 до 63.

Листинг 4

```

#include <iostream>

using namespace std;

int main()
{
    const int n = 64;
    int numbers[] {0, 1, 6, 5, 7, 2, 3, 63, 30, 21, 25};
    unsigned long long sorted = 0;
    unsigned long long mask = 1;

    for (auto i : numbers) {
        sorted |= mask << (n - i - 1);
    }

    for (int i = 0; i < n; i++) {
        if ((sorted & (mask << (n - i - 1))) > 0) {
            cout << i << ", ";
        }
    }

    return 0;
}

```

На листинге 5 показана программа сортировки с помощью битового массива, где в виде такого массива представлен вектор из чисел тип `unsigned char`. Если число, делённое на цело на 8 больше размера вектора в него добавляются новые элементы. После в нужную «ячейку» (элемент массива) записывается 1 в бит, равный порядковому номеру данного числа.

Листинг 5

```

#include <iostream>
#include <bitset>
#include <vector>

using namespace std;

int main()
{
    const int n = 64;

```

```

int numbers[] {0, 1, 6, 5, 7, 2, 3, 63, 30, 21, 25};
unsigned long long mask = 1;
vector<unsigned char> sort_vec {0};

for (auto i : numbers) {
    if (sort_vec.size() < i / 8 + 1) {
        int size = sort_vec.size();
        for (int j = 0; j < i / 8 + 1 - size; j++) {
            sort_vec.push_back(0);
        }
    }
    sort_vec[i / 8] |= mask << (7 - i % 8);
}

for (int i = 0; i < sort_vec.size(); i++) {
    for (int j = 0; j < 8; j++) {
        if ((sort_vec[i] & (mask << (7 - j))) > 0) {
            cout << 8 * i + j << ", ";
        }
    }
}

return 0;
}

```

На рисунке ниже представлен результат работы сортировки, где сначала идёт сортировка с помощью битового массива в виде числа типа unsigned long long, а после сортировка битовой последовательностью в виде вектора.

```

0, 1, 2, 3, 5, 6, 7, 21, 25, 30, 63,
0, 1, 2, 3, 5, 6, 7, 21, 25, 30, 63,
Process returned 0 (0x0) execution time : 0.031 s
Press any key to continue.

```

Рисунок 3 - Результат работы сортировки

### Задание 3

Сортировка (листинг 6) реализована по алгоритму из задания 2 с добавлением ввода и вывода через файловые потоки. Так же добавлен подсчет времени выполнения алгоритма с помощью библиотеки ctime и подсчёт количества занимаемой битовым массивом оперативной памяти.

```

#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <ctime>

using namespace std;

```

```

int main()
{
    setlocale(LC_ALL, "ru");
    unsigned int start = clock();
    ifstream in("in.txt");
    ofstream out("out.txt");

    string line;
    unsigned long long mask = 1;
    vector<unsigned char> sort_vec {0};

    while (getline(in, line)) {
        unsigned long long i = stoi(line);
        if (sort_vec.size() < i / 8 + 1) {
            int size = sort_vec.size();
            for (int j = 0; j < i / 8 + 1 - size; j++) {
                sort_vec.push_back(0);
            }
        }
        sort_vec[i / 8] |= mask << (7 - i % 8);
    }


    for (int i = 0; i < sort_vec.size(); i++) {
        for (int j = 0; j < 8; j++) {
            if ((sort_vec[i] & (mask << (7 - j))) > 0) {
                out << 8 * i + j << endl;
            }
        }
    }

    cout << (clock() - start) / 1000.0 << " секунд" << endl;
    cout << sizeof(unsigned char) * sort_vec.size() / 1024.0 / 1024.0 << "
Мегабайт";

    return 0;
}


```

На рисунках ниже предоставлены входные и выходные файлы программы

 in.txt – Блокнот

Файл	Правка	Формат	Вид	Справка
1				
2				
9				
23				
160				
3				
6				
10				
15				
16				
19				

Рисунок 4 - Входной файл

 out.txt – Блокнот

Файл	Правка	Формат	Вид	Справка
1				
2				
3				
6				
9				
10				
15				
16				
19				
23				
160				

Рисунок 5 - Выходной файл

Оперативная память, которую будет занимать битовый массив зависит от количества в нём «ячеек», а количество «ячеек» в свою очередь зависит от чисел во входном файле. Объём оперативной памяти вычисляется с помощью умножения количества элементов в массиве (`sort_vec.size()`) на вес одного такого элемента (так как все элементы массива являются числами типа `unsigned char`, то вес одного такого числа будет равен 1 байту)

## **Вывод**

Поставленные задачи выполнены, изучено битовое представление целых чисел и битовые операции в C++, реализованы и протестированы сортировки с помощью битового массива.

## **Литература**

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 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).