



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

Таганов А.А.

Практическую работу выполнил

«__» 2025 г.

«Зачтено»

«__» 2025 г.

Москва 2025

Задание №1

Реализовал пример, приведенный в практике, и проверил правильность результата на еще одном значении x.

Листинг1

```
#include <iostream>

using namespace std;

int main()
{
    unsigned char x = 255; // 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; // должно быть 255 - 16 = 239

    x = 17; // 8-разрядное двоичное число 00010001 (17)
    x &= (~mask << 4)); // 5 бит в 0
    cout << (int)x << endl; // должно быть 1
    return 0;
}
```

```
239
1

Process returned 0 (0x0)  execution time : 0.100 s
Press any key to continue.
```

Рис.1 – Вывод программы

Реализовал по аналогии с предыдущим примером установку 7 бита в единицу.

Листинг 2

```
#include <iostream>

using namespace std;

int main()
{
    unsigned char x = 17;
    unsigned char mask = 1;
    x |= mask << 6; // 7 бит в 1
    cout << (int)x << endl; // 17 + 64 = 81
    return 0;
}
```

81

```
Process returned 0 (0x0) execution time : 0.135 s
Press any key to continue.
```

Рис.2 – Вывод программы

Реализовал код из листинга 1, представленного в практике.

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: 00000000000000000000000000000001100  
Для продолжения нажмите любую клавишу . . . |
```

Рис.3 – Вывод программы

Данная программа выводит двоичное представление числа 25 в виде 32 разрядного числа. Каждый проход цикла выполняется вывод результата операции и с маской и сдвигение на $n - i$, затем маска сдвигается на 1 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 меняется на единицу.

Листинг 3

```
#include <iostream>
#include <bitset>
using namespace std;

int main() {
    const int maxNum = 8;
    unsigned char bitArray = 0;
    int count;

    cout << "How much numbers? ";
    cin >> count;
    if (count > maxNum) {
        cout << "Oops, too much";
        return 1;
    }

    for (int i = 0; i < count; i++) {
        int num;
        cout << "Enter your number(0-7)" << endl;
        cin >> num;
        bitArray = bitArray & ~(1 << (7 - i));
        bitArray = bitArray | (1 << (7 - i)) * (num == 1);
    }
}
```

```

if (num > 7) {
    cout << "Sorry, can't do" << endl;
    return 1;
}

bitArray |= (1 << (maxNum - 1 - num));
}

cout << "bitArray: " << bitset<8>(bitArray) << endl;

cout << "Sorted: ";
for (int i = 0; i < maxNum; i++) {
    if (bitArray & (1 << (maxNum - 1 - i))) {
        cout << i << " ";
    }
}
cout << endl;

return 0;
}

```

```

How much numbers? 5
Enter your number(0-7)
5
Enter your number(0-7)
2
Enter your number(0-7)
3
Enter your number(0-7)
0
Enter your number(0-7)
1
bitArray: 11110100
Sorted: 0 1 2 3 5

Process returned 0 (0x0) execution time : 9.652 s
Press any key to continue.

```

Рис.4 – Вывод программы

В листинге 4 представлен вариант кода, поддерживающий сортировку в диапазоне от 0 до 63.

```

#include <iostream>
#include <bitset>
using namespace std;

int main() {
    const int maxNum = 64;
    unsigned long long bitArray = 0;
    int count;

    cout << "How much numbers? ";
    cin >> count;
    if (count > maxNum) {
        cout << "Oops, too much";
        return 1;
    }

    for (int i = 0; i < count; i++) {
        int num;
        cout << "Enter your number(0-63)" << endl;
        cin >> num;
        if (num > maxNum - 1) {
            cout << "Sorry, can't do" << endl;
            return 1;
        }
        bitArray |= (1ULL << (maxNum - 1 - num));
    }

    cout << "bitArray: " << bitset<64>(bitArray) << endl;

    cout << "Sorted: ";
    for (int i = 0; i < maxNum; i++) {
        if (bitArray & (1ULL << (maxNum - 1 - i))) {
            cout << i << " ";
        }
    }
    cout << endl;

    return 0;
}

```

Рис.5 – Вывод программы

На листинге 5 представлена доработанная программа, где битовый массив представлен в виде вектора из чисел типа `unsigned char`.

Листинг 5

```
#include <iostream>
#include <bitset>
#include <vector>
using namespace std;

int main() {
    vector<unsigned char> bitArray {0};
    int count;

    cout << "How much numbers? ";
    cin >> count;

    for (int i = 0; i < count; i++) {
        int num;
        cout << "Enter your number" << endl;
        cin >> num;
        if (bitArray.size() < num / 8 + 1) {
            int size = bitArray.size();
            for (int j = 0; j < num / 8 + 1 - size; j++)
                bitArray.push_back(0);
        }
        bitArray[num / 8] |= (1 << (7 - (num % 8)));
    }
}
```

```

cout << "bitArray: ";
for (int i = 0; i < bitArray.size(); i++) {
    cout << bitset<8>(bitArray[i]) << " ";
}
cout << endl;

cout << "Sorted: ";
for (int i = 0; i < bitArray.size(); i++) {
    for (int j = 0; j < 8; j++) {
        if (bitArray[i] & (1 << (7 - j))) {
            cout << i * 8 + j << " ";
        }
    }
}
return 0;
}

```

```

How much numbers? 5
Enter your number
68
Enter your number
3
Enter your number
2
Enter your number
4
Enter your number
1
bitArray: 01111000 00000000 00000000 00000000 00000000 00000000 00000000 00001000
Sorted: 1 2 3 4 68
Process returned 0 (0x0)   execution time : 12.008 s
Press any key to continue.

```

Рис.6 – Вывод программы

Задание 3

На листинге 6 представлена сортировка по алгоритму из задачи 2. Было добавлено чтение заранее созданного текстового файла, вывод в новый текстовый файл и подсчет времени выполнения алгоритма с помощью библиотеки `ctime`.

Листинг 6

```

#include <iostream>
#include <bitset>
#include <vector>
#include <fstream>
#include <ctime>
using namespace std;

int main() {
    unsigned int start = clock();
    ifstream in("input.txt");
    ofstream out("output.txt");
    vector<unsigned char> bitArray {0};
    string line;

    while (getline(in, line)) {
        unsigned long long i = stoi(line);
        if (bitArray.size() < i / 8 + 1) {
            int size = bitArray.size();

```

```

        for (int j = 0; j < i / 8 + 1 - size; j++) {
            bitArray.push_back(0);
        }
    }

    bitArray[i / 8] |= (1 << (7 - (i % 8)));
}

for (int i = 0; i < bitArray.size(); i++) {
    for (int j = 0; j < 8; j++) {
        if (bitArray[i] & (1 << (7 - j))) {
            out << i * 8 + j << endl;
        }
    }
}
cout << (clock() - start) / 1000.0 << " seconds" << endl;
cout << sizeof(unsigned char) * bitArray.size() / 1024.0 / 1024.0 << " MByte";

return 0;
}

```

```

0.001 seconds
2.57492e-05 MByte
Process returned 0 (0x0) execution time : 0.112 s
Press any key to continue.

```

Рис.7 – Вывод программы

```

7
12
6
5
132
24
7
160
99
210
54
53
52|

```

Рис.8 – Исходный текстовый файл

```
|5  
6  
7  
12  
24  
52  
53  
54  
99  
132  
160  
210
```

Рис.9 – Отсортированный текстовый файл

Вывод

Поставленные задачи выполнены, изучены 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/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).