



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
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)

**ОТЧЕТ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ №5.1 по
дисциплине**

«Структуры и алгоритмы обработки данных»

Тема: Работа с данными из файла

Выполнил студент группы ИКБО-41-23

Гольд Д.В.

Принял старший преподаватель

Рысин М.Л.

Москва 2024

Оглавление

1.ЦЕЛЬ РАБОТЫ.....	3
2.ХОД РАБОТЫ	3

1.ЦЕЛЬ РАБОТЫ

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

2.ХОД РАБОТЫ

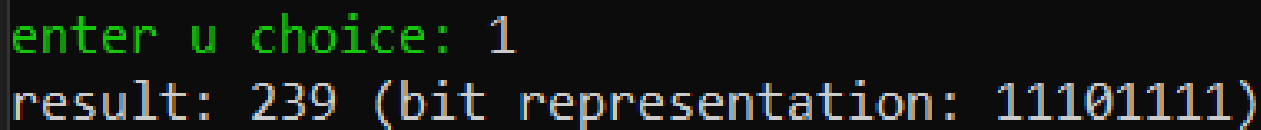
1.а: Реализация установки 5-й бит произвольного целого числа в 0, проверка правильности результата в том числе и на других значениях x. (Листинг 1.)

Листинг 1.

```
#include <iostream>
using namespace std; int
main() {
    unsigned char x = 255; // 11111111
    unsigned char maska = 1; // 00000001
    // установка 5ого бита (4-я позиция) в 0
    x = x & ~(maska << 4); // 11111111 & 11101111 = 11101111 (239)

    cout << "result: " << static_cast<int>(x) << " (bit representation: " <<
    bitset<8>(x) << ")" << endl;
}
```

Скрин работы программы при x = 255



```
enter u choice: 1
result: 239 (bit representation: 11101111)
```

1.6: Реализация по аналогии с предыдущим примером установку 7-го бита числа в единицу. (Листинг 2.)

Листинг 2.

```
unsigned char x = 175; // 10101111
unsigned char maska = 1; // 00000001

// установка 7ого бита (6-я позиция) в 1
x = x | (maska << 6); // 10101111 | 01000000 = 11101111 (239)

cout << "result: " << static_cast<int>(x) << " (bit representation: " << bitset<8>(x)
<< ")" << endl;
return;
```

Скрин работы программы при x = 175

```
enter u choice: 2
result: 239 (bit representation: 11101111)
```

1.в: Реализуйте код листинга 3, объяснение выводимого программой результата.

Листинг 3.

```
SetConsoleCP(1251);
SetConsoleOutputCP(1251);

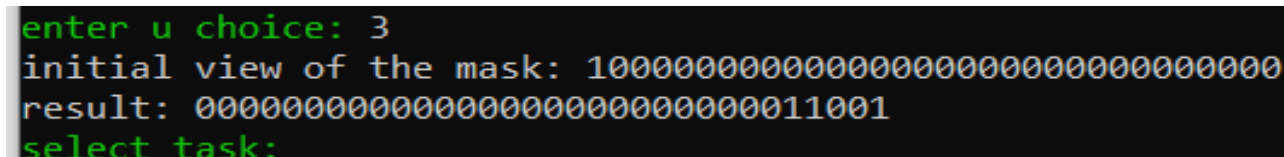
unsigned int x = 25;
const int n = sizeof(int) * 8; // 32 бита для типа int
unsigned int maska = (1 << (n - 1)); // 1 в старшем бите 32-разрядного числа

cout << "initial view of the mask: " << bitset<n>(maska) << endl; // вывод маски:
10000000000000000000000000000000

cout << "result: ";
for (int i = 1; i <= n; i++) {
    cout << ((x & maska) >> (n - i)); // проверка каждого бита
    maska = maska >> 1; // сдвигаем маску вправо на 1 бит
}

cout << endl;
return;
```

Скрин работы программы



```
enter u choice: 3
initial view of the mask: 10000000000000000000000000000000
result: 000000000000000000000000000011001
select task:
```

Программа на C++ преобразует число типа int (в данном случае 25) в двоичное представление. Она создает маску для проверки каждого бита числа и выводит их значения от старшего к младшему. В конце программа ожидает нажатия клавиши перед завершением.

2.а: Реализуйте ввода произвольного набора до 8-ми чисел (со значениями от 0 до 7) и его сортировкой битовым массивом в виде числа типа unsigned char. Проверка работу программы. (Листинг 4)

Листинг 4.

```

unsigned char array_bit = 0;
int number_digits, number;

cout << "enter the number of numbers (no more than 8): ";

while (!(cin >> number_digits) || number_digits > 8 || number_digits < 1) {
    SetConsoleTextAttribute(back_col, 0x0C);
    cout << "error!!! enter a valid number (1 to 8): ";
    SetConsoleTextAttribute(back_col, 0x07);
    cin.clear();
    cin.ignore();
}

cout << "enter a number from 0 to 7: ";
for (int i = 0; i < number_digits; i++) {
    while (!(cin >> number) || number < 0 || number > 7) {
        SetConsoleTextAttribute(back_col, 0x0C);
        cout << "error!!! the numbers must be in the range from 0 to 7: ";
        SetConsoleTextAttribute(back_col, 0x07);
        cin.clear();
        cin.ignore();
    }
    array_bit |= (1 << number); // установка соответствующего бита
}

cout << "\nsorted numbers: ";
for (int i = 0; i < 8; i++) {
    if (array_bit & (1 << i)) { // проверка каждого бита
        cout << i << " ";
    }
}
cout << endl << endl;

```

Скрин работы программы

```

enter u choice: 4
enter the number of numbers (no more than 8): 6
enter a number from 0 to 7: 0
1
2
3
4
6

sorted numbers: 0 1 2 3 4 6

```

2.б: Адаптируйте задание 2.а для набора из 64-х чисел (со значениями от 0 до 63) с битовым массивом в виде числа типа unsigned long long. (Листинг 5)

Листинг 5

```
unsigned long long array_bit = 0;
int number_digits, number;

cout << "enter the number of numbers (no more than 64): ";
cin >> number_digits;

if (number_digits > 64) {
    cout << "error!!! the number of numbers must not exceed 64!" << endl;

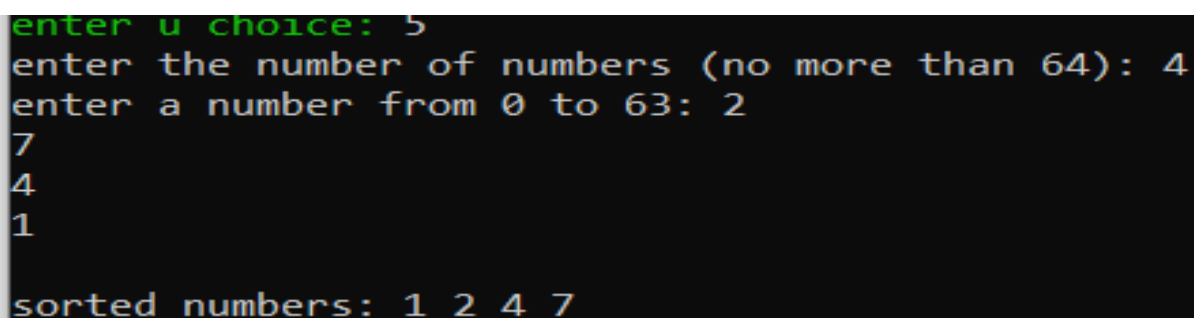
    return;
}

cout << "enter a number from 0 to 63: ";
for (int i = 0; i < number_digits; i++) {
    cin >> number;
    if (number < 0 || number > 63) {
        cout << "error!!! the numbers must be in the range from 0 to 63!" << endl;
        return;
    }
    array_bit |= (1ULL << number); // установка соответствующего бита
}

cout << "\nsorted numbers: ";
for (int i = 0; i < 64; i++) {
    if (array_bit & (1ULL << i)) { // проверка каждого бита
        cout << i << " ";
    }
}
cout << endl << endl;

return;
```

Скрин работы программы



```
enter a choice: 5
enter the number of numbers (no more than 64): 4
enter a number from 0 to 63: 2
7
4
1

sorted numbers: 1 2 4 7
```

2.в: Исправление программы задания 2.б, чтобы для сортировки набора из 64х чисел использовалось не одно число типа unsigned long long, а линейный массив чисел типа unsigned char. (Листинг 6.)

Листинг 6.

```
const int number_bytes = 8; // 64 числа по 8 бит на байт
unsigned char array_bit[number_bytes] = { 0 };
int number_digits, number;

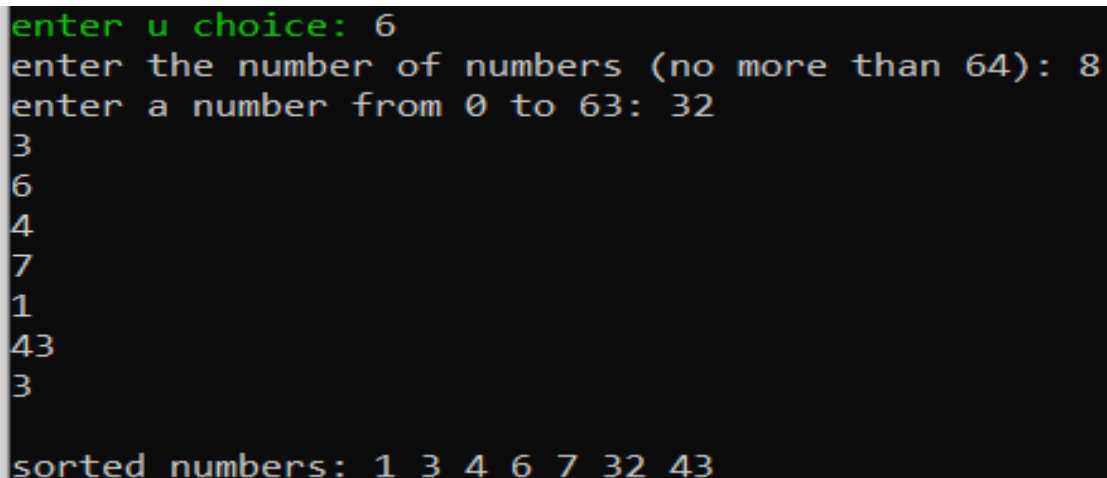
cout << "enter the number of numbers (no more than 64): ";
cin >> number_digits;

if (number_digits > 64) {
    cout << "error!!! the number of numbers must not exceed 64" << endl;
    return;
}

// установка соответствующих битов
cout << "enter a number from 0 to 63: ";
for (int i = 0; i < number_digits; i++) {
    cin >> number;
    if (number < 0 || number > 63) {
        cout << "error!!! the numbers must be in the range from 0 to 63!" << endl;
        return;
    }
    array_bit[number / 8] |= (1 << (number % 8)); // установка соответствующего бита
}

// выводим индексы единичных битов
cout << "\nsorted numbers: ";

for (int i = 0; i < 64; i++) {
    if (array_bit[i / 8] & (1 << (i % 8))) { // проходимся по каждому биту
        cout << i << " ";
    }
}
cout << endl;
```



The screenshot shows the program's execution. It prompts the user to enter a choice, the number of numbers (up to 64), and then 8 numbers. The numbers entered are 3, 6, 4, 7, 1, 43, 3, and 32. The program then outputs the sorted numbers: 1 3 4 6 7 32 43.

```
enter u choice: 6
enter the number of numbers (no more than 64): 8
enter a number from 0 to 63: 3
6
4
7
1
43
3
32

sorted numbers: 1 3 4 6 7 32 43
```


3.a: Реализация задачи сортировки числового файла с заданными условиями и добавление возможности определения времени работы программы. (Листинг 7)

Листинг 7

```
void three_task() {
    int number_of_unique_numbers = 10000000;
    fill_input_file_with_unique_numbers(INPUT_FILE, number_of_unique_numbers); //
    заполняем входной файл

    clock_t t0 = clock();

    // размер битового массива: делим на 8, чтобы вместить биты в char
    vector<char> ArrayBit(MAX_NUM / 8, 0); // инициализация значениями 0

    ifstream inputFile(INPUT_FILE);
    if (!inputFile.is_open()) {
        cerr << "error open file" << INPUT_FILE << endl;
        return;
    }

    int number;
    while (inputFile >> number) {
        if (number >= 0 && number < MAX_NUM) {
            // устанавливаем соответствующий бит в 1
            ArrayBit[number / 8] |= (1 << (number % 8)); // находим правильный байт
            и устанавливаем бит
        }
    }
    inputFile.close();

    ofstream outputFile(OUTPUT_FILE);
    if (!outputFile.is_open()) {
        cerr << "error open file" << OUTPUT_FILE << endl;
        return;
    }

    // проходим по массиву и записываем индексы установленных битов в выходной файл
    for (int i = 0; i < MAX_NUM; ++i) {
        if (ArrayBit[i / 8] & (1 << (i % 8))) { // проверяем, установлен ли бит
            outputFile << i << endl; // запись уникального числа
        }
    }
    outputFile.close();

    clock_t t1 = clock();

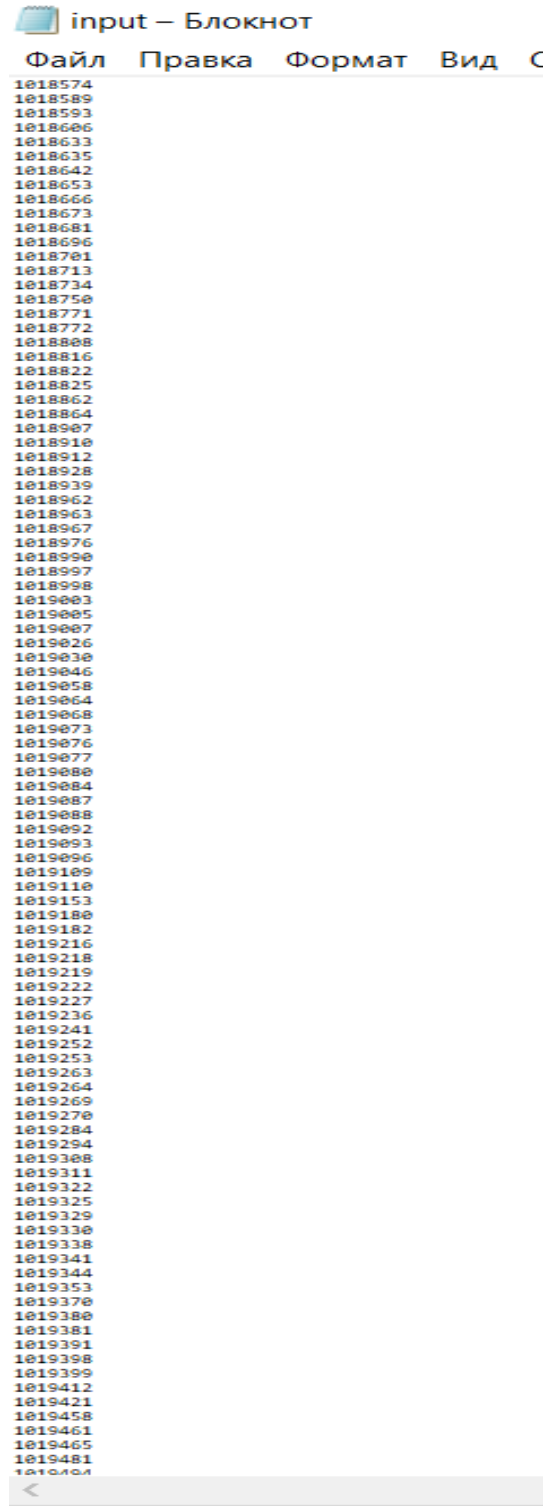
    cout << "numbers from the file " << INPUT_FILE << " sorted and placed in " <<
    OUTPUT_FILE << endl;
    cout << "program execution time: " << (double)(t1 - t0) / CLOCKS_PER_SEC << "
    seconds" << endl;

    return;
}
```

Скрин работы программы

```
enter u choice: 7
enter file 'input.txt' successfully filled with 1000000 unique numbers.
numbers from the file input.txt sorted and placed in output.txt
program execution time: 3.765 seconds
```

Файл с отсортированными числами



3.б:Определить программно объём оперативной памяти, занимаемый битовым массивом. (Листинг 8.)

Листинг 8.

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

const int MAX_NUM = 10000000;

int main() {
    vector<bool> ArrayBit(MAX_NUM,
0);

    size_t memorySize = ArrayBit.capacity() / 8; // так как в vector<bool> каждый
элемент занимает 1 бит

    cout << "the amount of memory occupied by a bit array: " << memory Size << " byte" <<
endl; cout << "the amount of memory in megabytes: " <<
static_cast<double>(memorySize) / (1024 * 1024) << " MB" << endl;    return 0;
}
```

Скрин работы программы.

```
the amount of memory occupied by a bit array: 250000 byte
the amount of memory in megabytes: 1.82909 MB
```