

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 |  |  | «\_\_»\_\_\_\_\_\_\_2022 г. | \_\_\_\_\_\_\_\_\_\_ |
| работа выполнена |  |  |  |  |
| «Зачтено» |  |  | «\_\_»\_\_\_\_\_\_\_2022 г. | \_\_\_\_\_\_\_\_\_\_ |
|  |  |  | Москва 2022 |  |

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc113565358)

[1 ЦЕЛЬ РАБОТЫ 3](#_Toc113565359)

[2 ХОД РАБОТЫ 4](#_Toc113565360)

[2.1 Задание 1.а 4](#_Toc113565361)

[2.2 Задание 1.б 5](#_Toc113565362)

[2.3 Задание 1.в 6](#_Toc113565363)

[2.4 Задание 2.a 7](#_Toc113565364)

[2.5 Задание 2.б 8](#_Toc113565365)

[2.6 Задание 2.в 9](#_Toc113565366)

[2.7 Задание 3.а и 3.б 10](#_Toc113565367)

[3 ВЫВОД 13](#_Toc113565368)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#_Toc113565369)

# ЦЕЛЬ РАБОТЫ

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

# ХОД РАБОТЫ

## 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 смещения, 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 сохранят 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.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 - Результат тестирования на числе 255

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 - Результат тестирования на числе 16

## Задание 1.б

**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 б **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.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 - Результат тестирования на числе 1

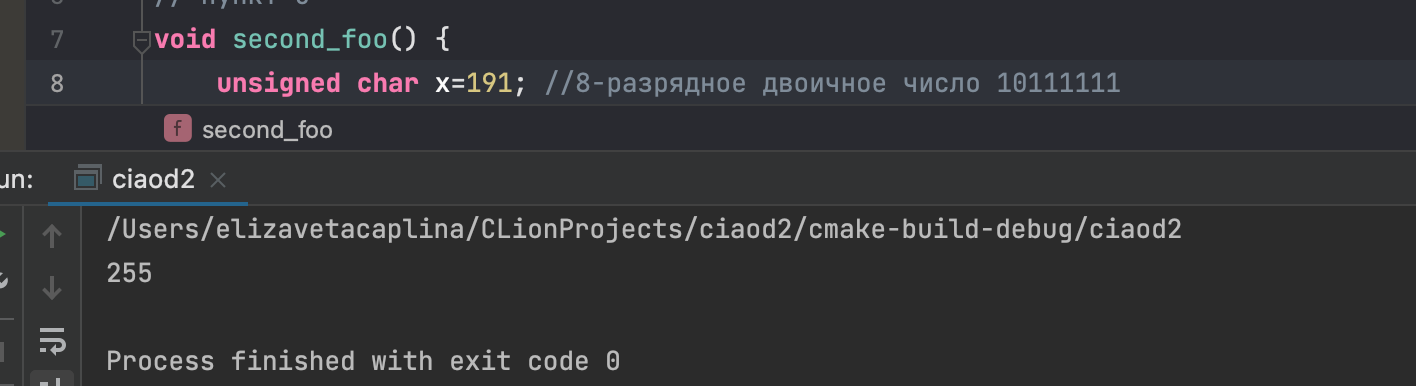


Рисунок 4 - Результат тестирования на числе 191

## 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.

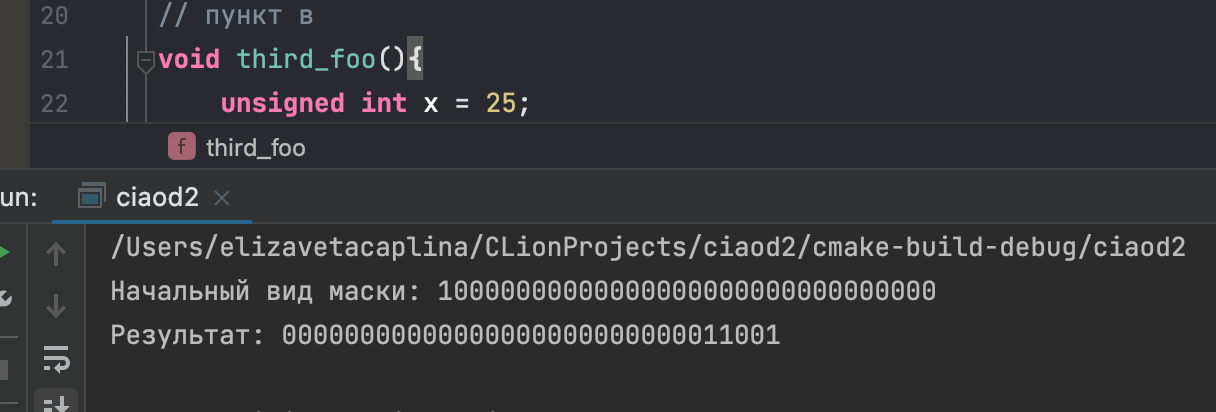


Рисунок 5 - Результат тестирования на числе 25

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 - Результат тестирования на числе 255

## Задание 2.a

**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.

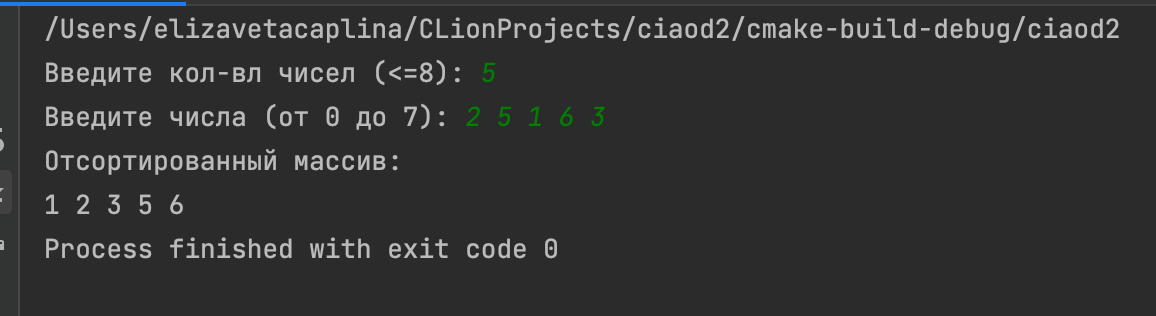


Рисунок 7 - Результат тестирования программы

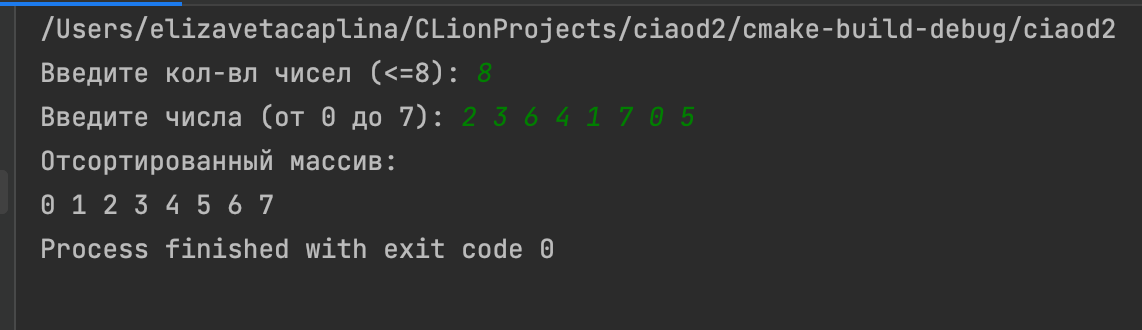


Рисунок 8 - Результат тестирования программы

## Задание 2.б

**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.

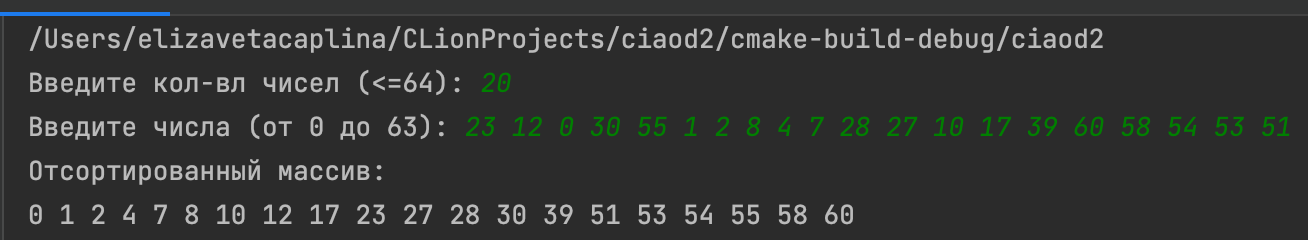


Рисунок 9 - Результат тестирования программы

## 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.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 - Результаты работы программы

## 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 рандомными 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.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11 - Результат работы программы

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 - Результат работы программы

# 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 С++ [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).