

МИНОБРНАУКИ РОСИИ

*Федеральное государственное бюджетное образовательное учреждение высшего образования*

***«МИРЭА – Российский технологический университет»***

**РТУ МИРЭА**



Отчет по выполнению практического задания №5.1

**Тема:**

Битовые операции. Сортировка числового файла с помощью битового массива.Дисциплина: «Структуры и алгоритмы обработки данных»

|  |  |  |
| --- | --- | --- |
| Выполнил студент: Васильев Б.А.  Группа: ИКБО-20-23 |  |  |

Москва 2024

**СОДЕРЖАНИЕ**

[ЗАДАНИЕ 1 3](#_Toc176887189)

[Формулировка задачи 3](#_Toc176887190)

[Математическая модель решения (описание алгоритма) 4](#_Toc176887191)

[Коды программ 5](#_Toc176887192)

[Результаты тестирования 7](#_Toc176887193)

[ЗАДАНИЕ 2 9](#_Toc176887194)

[Формулировка задачи 9](#_Toc176887195)

[Математическая модель решения (описание алгоритма) 9](#_Toc176887196)

[Коды программ 11](#_Toc176887197)

[Результаты тестирования 13](#_Toc176887198)

[ЗАДАНИЕ 3 15](#_Toc176887199)

[Формулировка задачи 15](#_Toc176887200)

[Математическая модель решения (описание алгоритма) 16](#_Toc176887201)

[Коды программ 17](#_Toc176887202)

[Результаты тестирования 18](#_Toc176887203)

[ВЫВОД 19](#_Toc176887204)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 19](#_Toc176887205)

**ЦЕЛЬ РАБОТЫ**

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

# ЗАДАНИЕ 1

### Формулировка задачи

Определить делится ли число на каждую из своих цифр.

**1.a.** Реализуйте нижеприведённый пример с рисунка 1, проверьте правильность результата в том числе и на других значениях х.

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

**1.в.** Реализуйте код с рисунка 2, объясните выводимый программой результат.

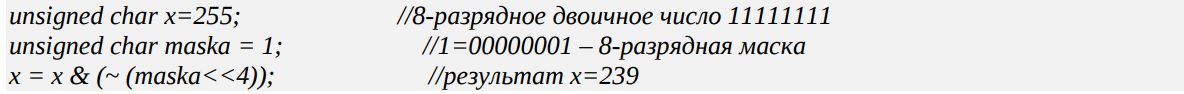


Рисунок 1 – Установка 5-го бита произвольного целого числа в 0

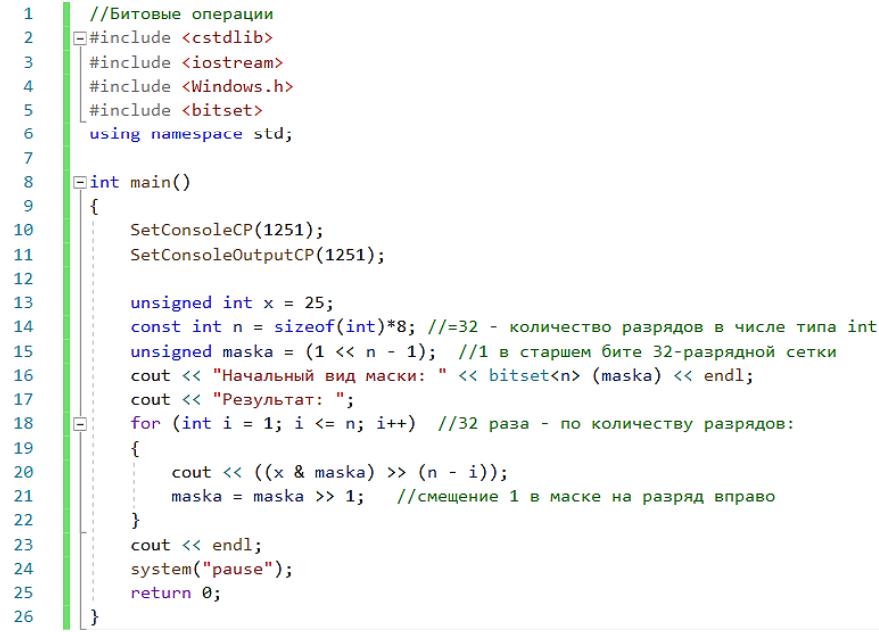


Рисунок 2 – к задаче 1.в

### Математическая модель решения (описание алгоритма)

В заданиях 1.а. и 1.б. происходит установка n-го бита целого числа в 0. Сначала у пользователя запрашивается целое число от 0 до 255 (максимальное число представимое восемью двоичными разрядами). Затем создаётся маска вида 00000001. Далее с помощью побитового сдвига влево на n разрядов, получаем маску с единицей в том разряде, в котором у исходного числа требуется выставить ноль, инвертируем маску и с помощью операции “побитовое И” получаем необходимый результат.

### Коды программ

Реализуем алгоритмы на языке программирования C++ (рис. 3-5)

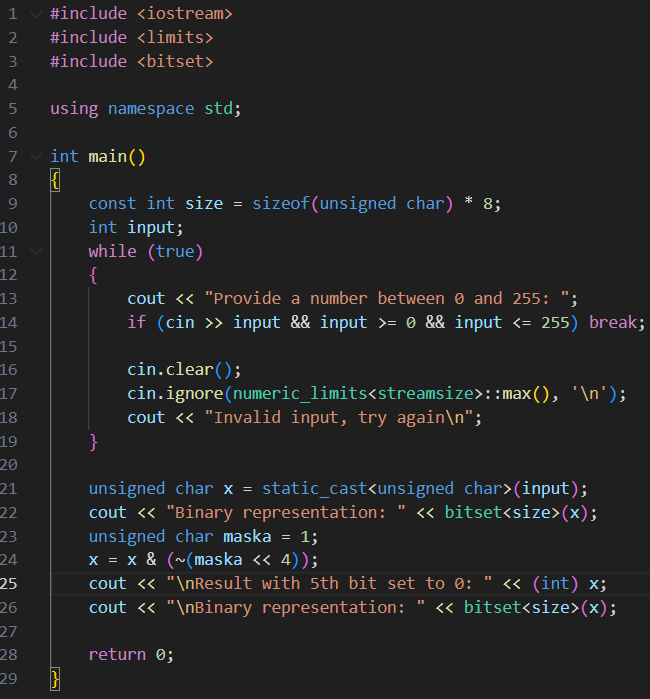


Рисунок 3 – Код к задаче 1.а

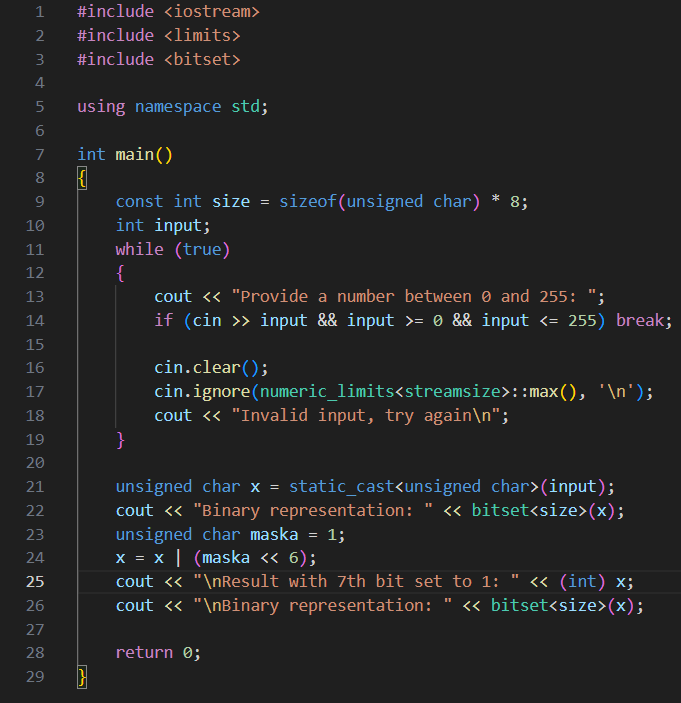


Рисунок 4 – Код к задаче 1.б

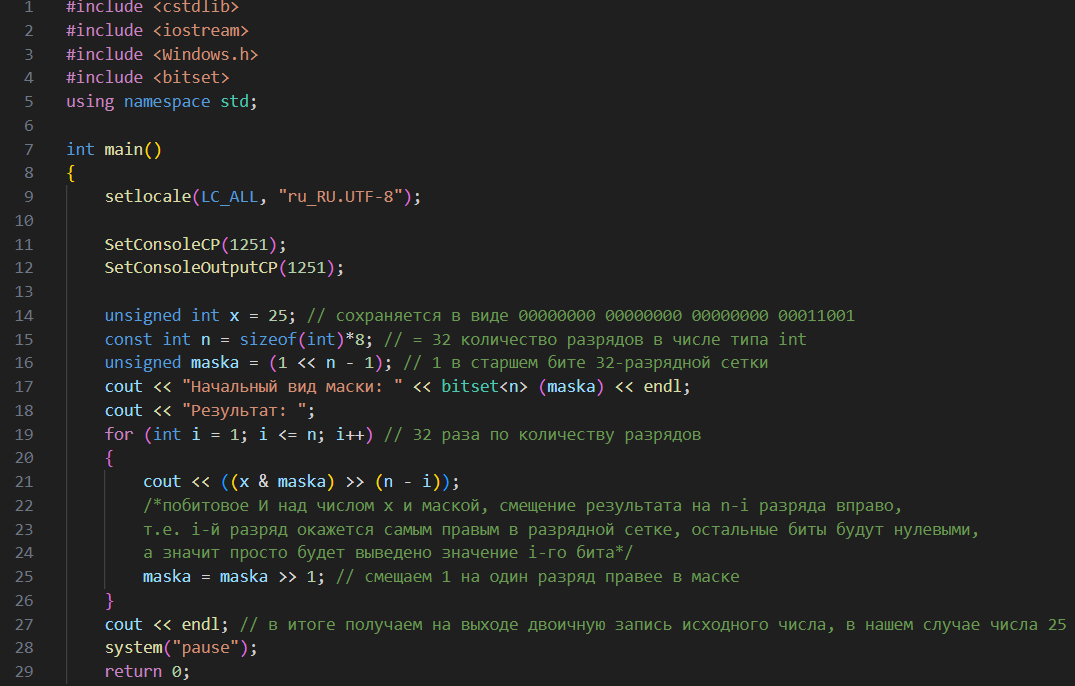


Рисунок 5 – Код к задаче 1.в и его объяснение

### Результаты тестирования

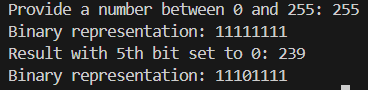
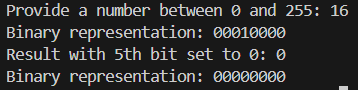
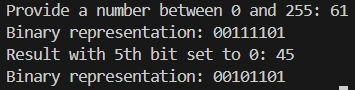


Рисунок 6 – Тестирование кода к задаче 1.а (часть 1)

  
Рисунок 7 – Тестирование кода к задаче 1.а (часть 2)

  
Рисунок 8 – Тестирование кода к задаче 1.а (часть 3)

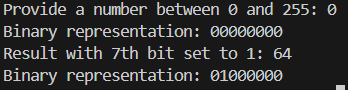
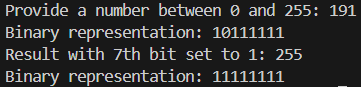
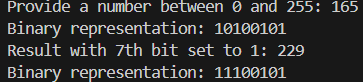
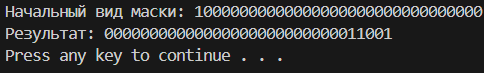


Рисунок 9 – Тестирование кода к задаче 1.б (часть 1)

  
Рисунок 10 – Тестирование кода к задаче 1.б (часть 2)

  
Рисунок 11 – Тестирование кода к задаче 1.б (часть 3)

  
Рисунок 12 – Тестирование кода к задаче 1.в

Тестирование показало, что программы работают исправно (рис. 6-12).

# ЗАДАНИЕ 2

### Формулировка задачи

Определить делится ли число на каждую из своих цифр.

**2.a.** Реализуйте сортировку последовательности чисел с помощью двоичного массива 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.

### Математическая модель решения (описание алгоритма)

Алгоритм решения задания 2.а:

1. **Ввод размера массива**:
   * Программа просит у пользователя ввести размер массива, который должен быть в пределах от 1 до 8 включительно.
   * Если введённое значение не является числом или находится вне указанного диапазона, программа выводит сообщение об ошибке и повторяет запрос.
2. **Ввод уникальных чисел**:
   * После успешного ввода размера массива программа запрашивает у пользователя ввести уникальные числа в диапазоне от 0 до 7.
   * Если введённое число не соответствует условиям (например, не число или вне диапазона), программа выводит сообщение об ошибке и повторяет запрос.
   * Введённые числа используются для установки соответствующих битов в переменной bit\_array.
3. **Установка битов**:
   * Программа использует побитовое ИЛИ с присвоением (|=) для установки битов в переменной bit\_array на основе введённых чисел. Единица из битовой маски перемещается влево на количество разрядов равное введённому числу, а затем идёт установка в единицу соответствующего разряда в переменной bit\_array.
4. **Вывод отсортированного массива**:
   * Программа проходит по всем 8 битам переменной bit\_array ,начиная с младшего, и проверяет, установлены ли они в единицу.
   * Если бит установлен, программа выводит соответствующее число.

Алгоритм решения задания 2.б аналогичен предыдущему:

1. **Ввод размера массива**:
   * Программа просит у пользователя ввести размер массива, который должен быть в пределах от 1 до 64 включительно.
   * Если введённое значение не является числом или находится вне указанного диапазона, программа выводит сообщение об ошибке и повторяет запрос.
2. **Ввод уникальных чисел**:
   * После успешного ввода размера массива программа запрашивает у пользователя ввести уникальные числа в диапазоне от 1 до 63.
   * Если введённое число не соответствует условиям (например, не число или вне диапазона), программа выводит сообщение об ошибке и повторяет запрос.
   * Введённые числа используются для установки соответствующих битов в переменной bit\_array.
3. **Установка битов**:
   * Программа использует побитовое ИЛИ с присвоением (|=) для установки битов в переменной bit\_array на основе введённых чисел. Единица из битовой маски перемещается влево на количество разрядов равное введённому числу, а затем идёт установка в единицу соответствующего разряда в переменной bit\_array.
4. **Вывод отсортированного массива**:
   * Программа проходит по всем 64 битам переменной bit\_array и проверяет, установлены ли они в единицу.
   * Если бит установлен, программа выводит соответствующее число.

Алгоритм решения задания 2.в:

1. **Инициализация данных:**
   * Создаётся вектор целых чисел numbers размером 64 для хранения вводимых значений.
   * Переменная max\_num используется для определения наибольшего числа из введённых.
2. **Ввод чисел**:
   * Программа просит пользователя ввести 64 числа.
   * Если очередное введённое число превосходит значение переменной max\_num, то происходит обновление значения этой переменной.
3. **Подготовка битового массива**:
   * Вычисляется количество байтов, необходимых для хранения битовой маски, используя наибольшее число из введённых. Это количество байтов хранится в bytes\_size.
   * Массив bytes инициализируется нулями. Размер массива bytes равен bytes\_size.
4. **Установка битов:**
   * Для каждого введённого числа определяется соответствующий байт и устанавливается соответствующий бит в этом байте. Это делается с помощью побитового ИЛИ с присвоением (|=) и сдвига маски (>>).
5. **Вывод отсортированной последовательности:**
   * Происходит проход по каждому биту каждого байта из массива bytes. Если очередной бит установлен в 1, то индекс этого бита из сквозной нумерации выводится в консоль.

### Коды программ

Реализуем алгоритмы на языке программирования C++ (рис. 13-16)

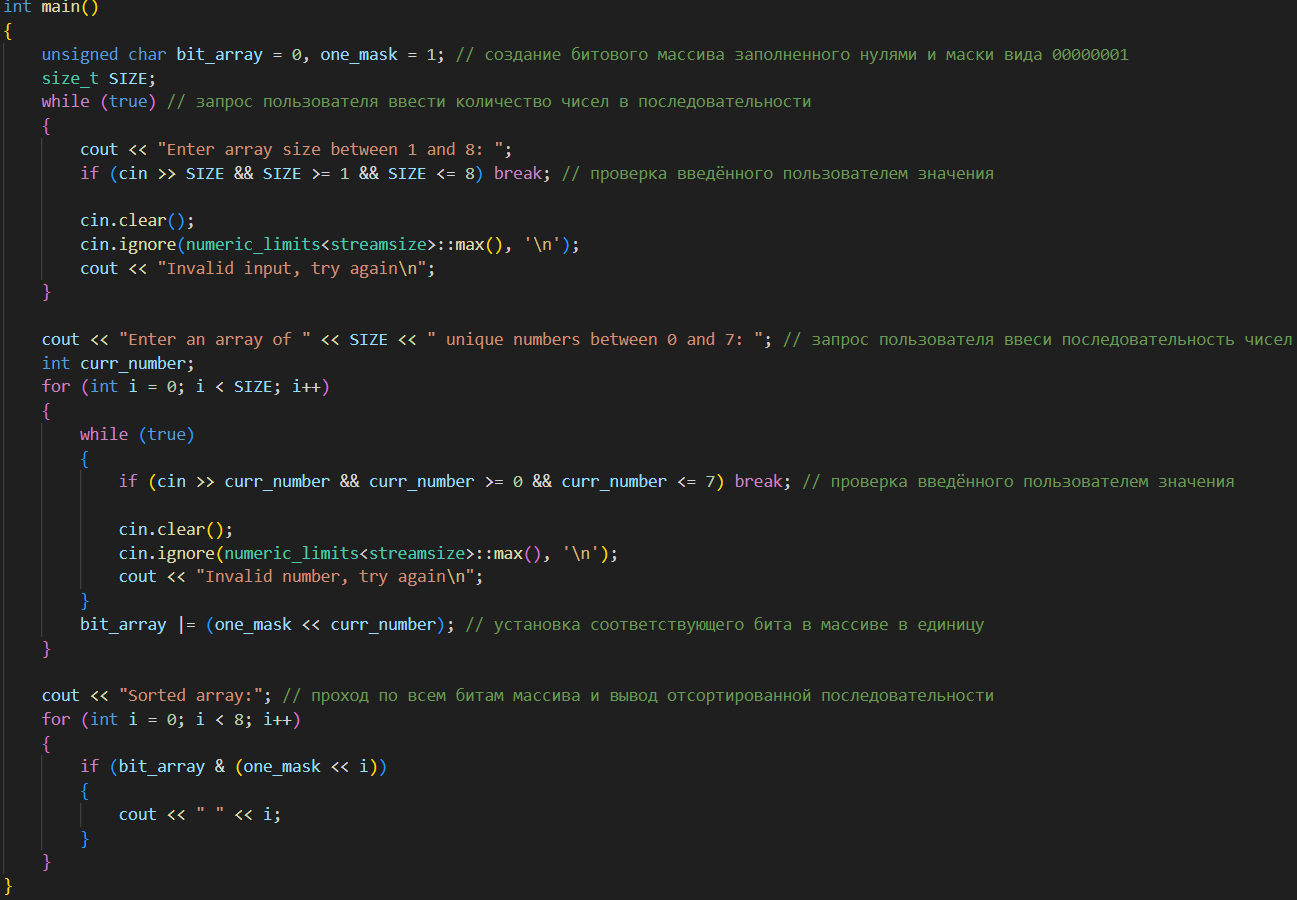


Рисунок 13 – Код к задаче 2.а

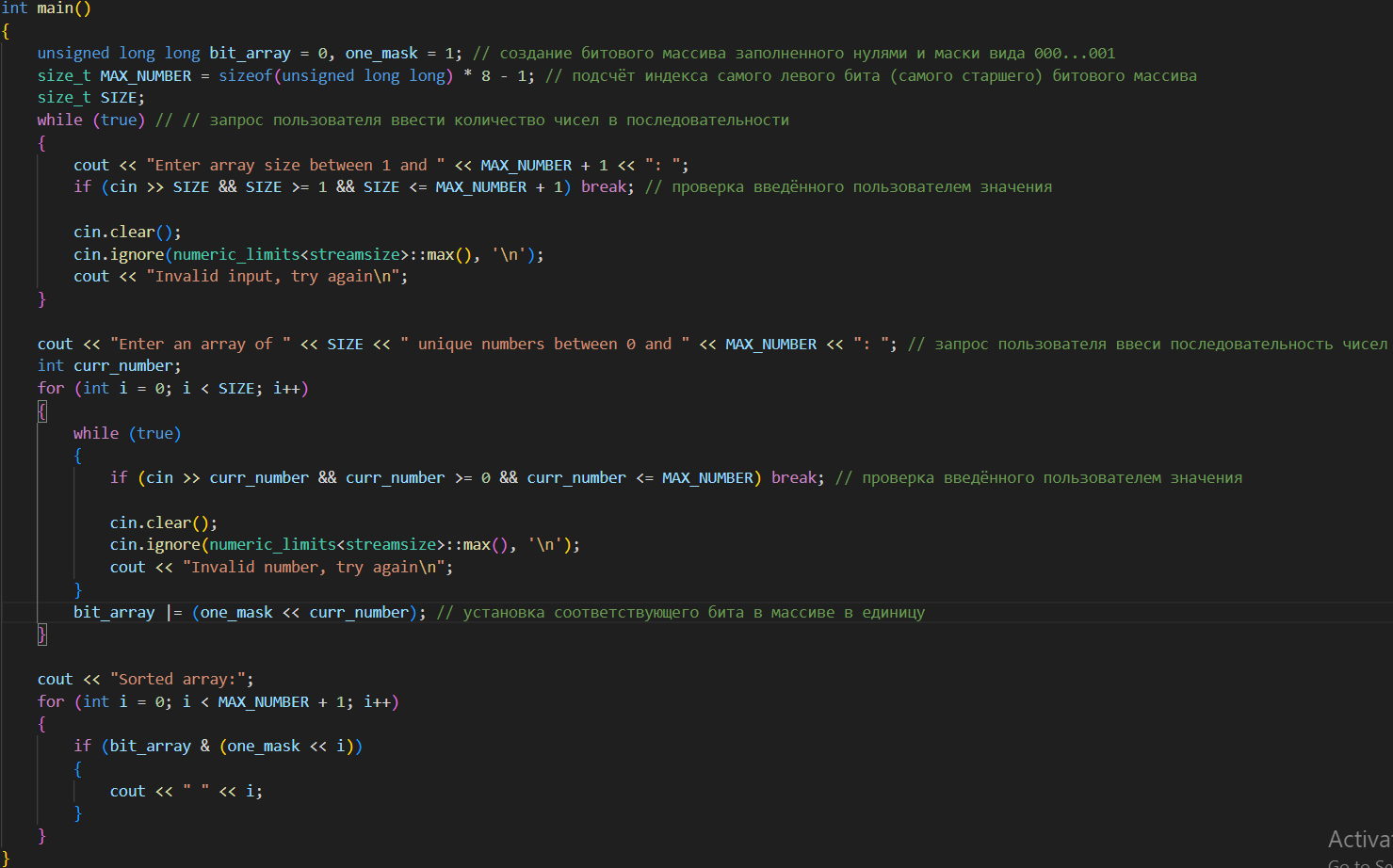


Рисунок 14 – Код к задаче 2.б

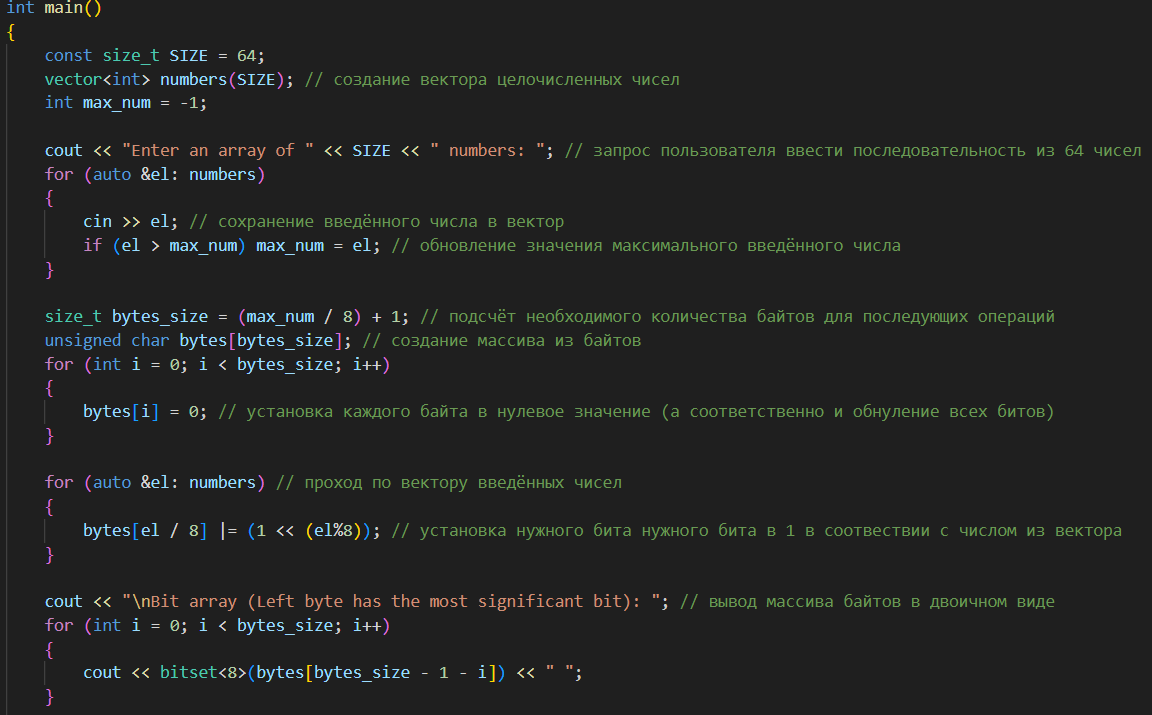


Рисунок 15 – Код к задаче 2.в (часть 1)

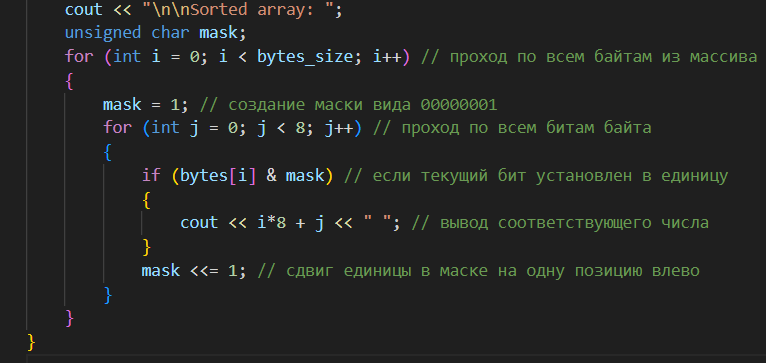


Рисунок 16 – Код к задаче 2.в (часть 2)

### Результаты тестирования

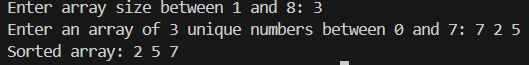
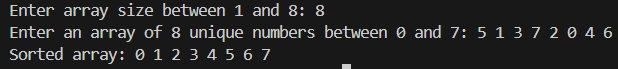
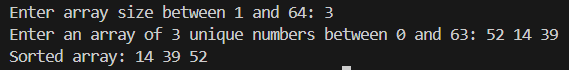


Рисунок 17 – Тестирование кода к задаче 2.а (часть 1)

  
Рисунок 18 – Тестирование кода к задаче 2.а (часть 2)

  
Рисунок 19 – Тестирование кода к задаче 2.б (часть 1)

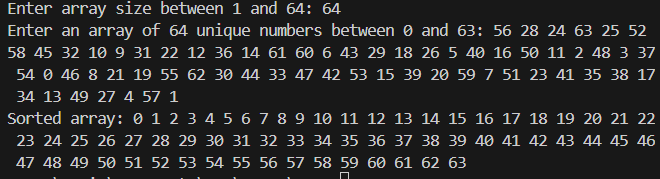
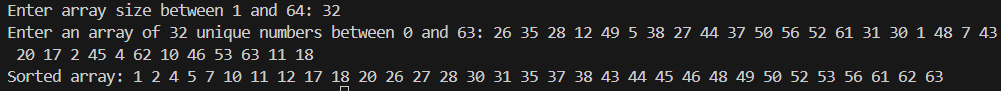
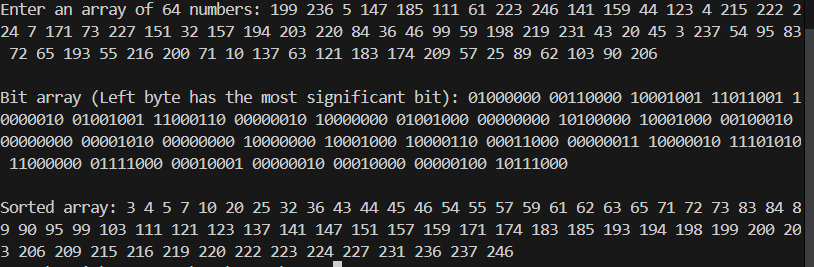
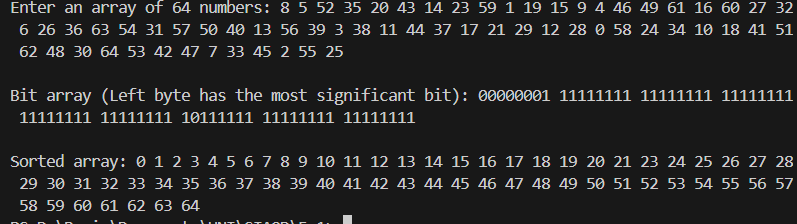


Рисунок 20 – Тестирование кода к задаче 2.б (часть 2)

  
Рисунок 21 – Тестирование кода к задаче 2.б (часть 3)

  
Рисунок 22 – Тестирование кода к задаче 2.в (часть 1)

  
Рисунок 23 – Тестирование кода к задаче 2.в (часть 2)

Тестирование показало, что программы работают исправно (рис.17-23).

# ЗАДАНИЕ 3

### Формулировка задачи

**Входные данные**: файл, содержащий не более n=107 неотрицательных целых чисел, среди них нет повторяющихся.

**Результат**: упорядоченная по возрастанию последовательность исходных чисел в выходном файле.

**Время работы программы**: ~10 с (до 1 мин. для систем малой вычислительной мощности).

**Максимально допустимый объём ОЗУ для хранения данных**: 1 МБ.

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

В отчёт внесите результаты тестирования для наибольшего количества входных чисел, соответствующего битовому массиву длиной 1МБ.

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

### Математическая модель решения (описание алгоритма)

Алгоритм решения задания 3.а и 3.б:

1. **Входные данные:**
   * Считываем набор целых чисел из файла. Необходимо определить, какие числа встречаются, и сохранить эту информацию в ОЗУ.
2. **Хранение чисел в виде битов**:
   * Используем динамический массив (вектор) байтов. Каждый байт в векторе содержит 8 битов, и каждый бит отвечает за информацию о наличии одного конкретного числа.
   * Размер вектора увеличивается по мере необходимости: если текущее число превышает количество уже доступных бит для его хранения, вектор автоматически расширяется, добавляя новые байты.
3. **Привязка чисел к битам**:
   * Для каждого числа N определяем, в каком байте и в каком бите этого байта оно должно храниться:
     1. Индекс нужного байта находим как целую часть от деления N на 8.
     2. Индекс нужного бита внутри байта находим как остаток от деления N на 8 .
   * Устанавливаем соответствующий бит в единицу с помощью битовой маски и операций побитового ИЛИ с присваиванием (|=) и сдвига (<<).
4. **Установка битов:**
   * Для каждого введённого числа определяется соответствующий байт и устанавливается соответствующий бит в этом байте. Это делается с помощью побитового ИЛИ с присвоением (|=) и сдвига маски (<<).
5. **Сохранение отсортированной последовательности:**
   * Закрываем файл со входными данным, он нам больше не потребуется
   * Открываем файл для записи.
   * Проходим последовательно по каждому биту каждого байта, начиная с самого младшего, если значение двоичного разряда равно 1, то записываем индекс этого бита в файл

### Коды программ

Реализуем алгоритм на языке программирования C++ (рис. 25-26)

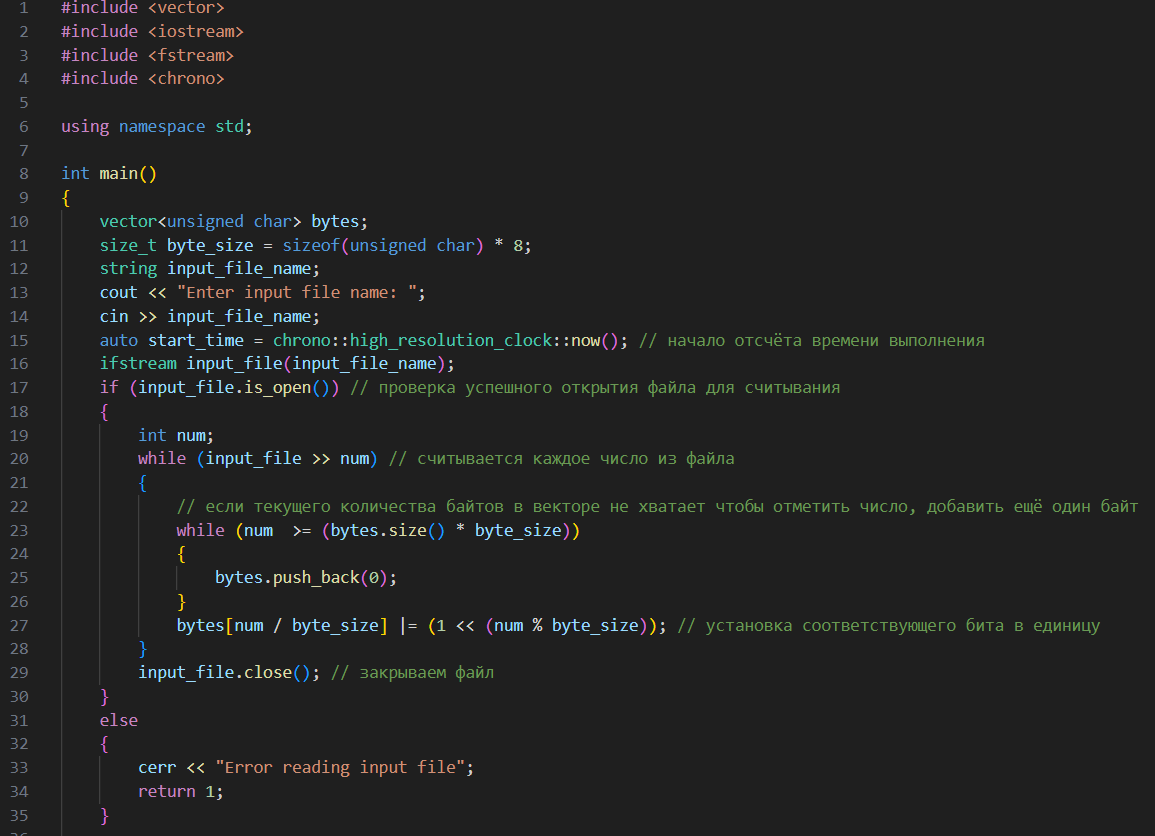


Рисунок 24 – Код к задаче 3.а и 3.б (часть 1)

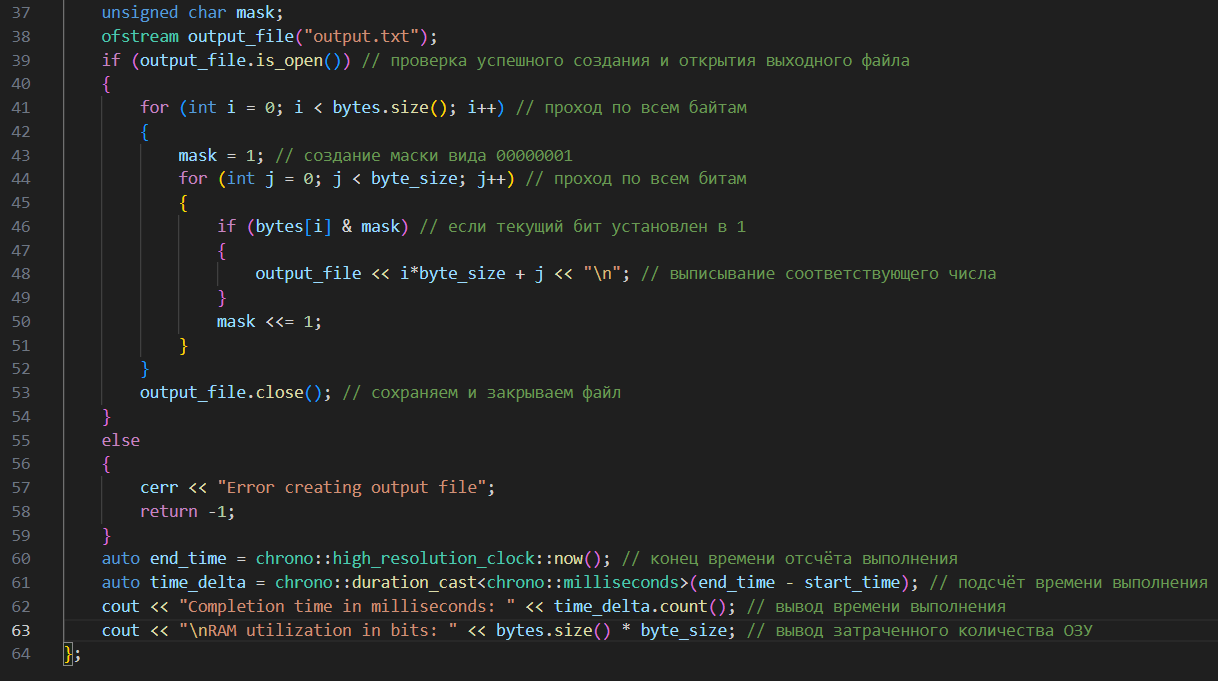


Рисунок 25 – Код к задаче 3.а и 3.б (часть 2)

### Результаты тестирования

Ограничение длины битового массива в 1 МБ позволяет обработать файл входных данных с неповторяющимися целыми неотрицательными числами, каждое из которых не превосходит по значению число 8 388 607, т.к 1 МБ = 1024\*1 КБ = 1024\*1024\*1 байт = 1024\*1024\*8 бит = 8 388 608 бит. Создадим файл из 8 388 608 чисел (рис. 27) и проверим нашу программу на нём (рис. 28).

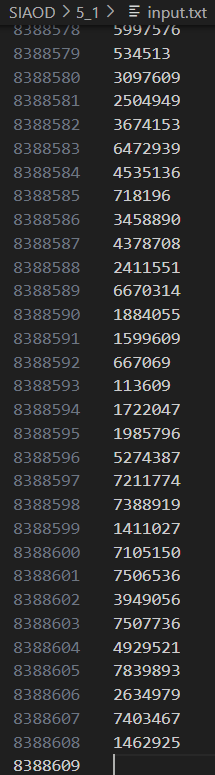
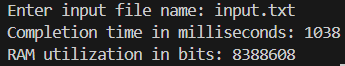


Рисунок 27 – Фрагмент файла входных данных

  
Рисунок 28 – Тестирование кода к задаче 3.а и 3.б

Как видно, программа работает корректно и укладывается в поставленные ограничения по времени работы и количеству используемой памяти.

# ВЫВОД

В результате выполнения работы освоены приёмы работы с битовым представлением беззнаковых целых чисел, а также реализован эффективный алгоритм внешней сортировки на основе битового массива.

# СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Рысин, М. Л. Введение в структуры и алгоритмы обработки данных : учебное пособие / М. Л. Рысин, М. В. Сартаков, М. Б. Туманова. — Москва : РТУ МИРЭА, 2022 — Часть 2 : Поиск в тексте. Нелинейные структуры данных. Кодирование информации. Алгоритмические стратегии — 2022. — 111 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/310826 (дата обращения: 10.09.2024).
2. Документация по языку С++ [Электронный ресурс]. URL: https://docs.microsoft.com/ruru/cpp/cpp/ (дата обращения 10.09.2024).