

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»

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-01-21 | Боронин Н.А |
|  |  |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_» cентября 2022 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | (*подпись студента*) |
| «Зачтено» | «\_\_» \_\_\_\_\_\_\_\_\_2022 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | *(подпись руководителя)* |

Москва 2022

1. **Цель работы**

Целью данной работы является освоение приёмов работы с битовым представлением беззнаковых целых чисел, и реализация эффективного алгоритма внешней сортировки на основе битового массива.

1. **Ход работы**
   1. **Задание 1**
      1. **Формулировка задачи**

Необходимо реализовать следующие программы:

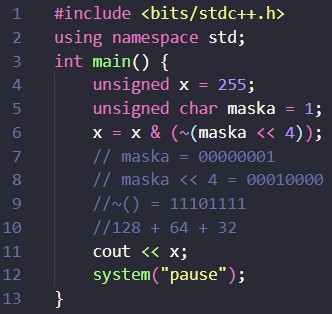
* приведенный в методических указаниях пример, проверить его правильность на других значениях переменной x;
* установка 7-го бита числа в единицу;
* приведенный в листинге 1 пример, объяснив выводимый программой результат.
  + 1. **Описание алгоритма**

Касаемо второй задачи: для того, чтобы установить 7-ой бит числа в единицу, необходимо применить побитовый сдвиг влево на шесть разрядов.

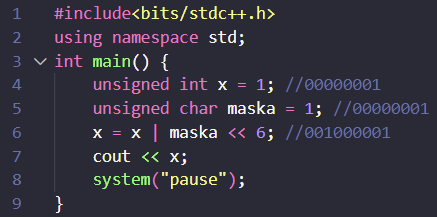
Третья программа работает таким образом: маской является 32-х разрядное число с единицей в старшем разряде. Далее в цикле на каждом i-том шаге после выполнения побитового И, производится побитовый сдвиг вправо таким образом, чтобы в младшем разряде оказался i-тый бит числа и выводится результат. После этого единица в маске смещается на один разряд вправо, таким образом, чтобы можно было вывести следующий бит числа.

* + 1. **Код программы**

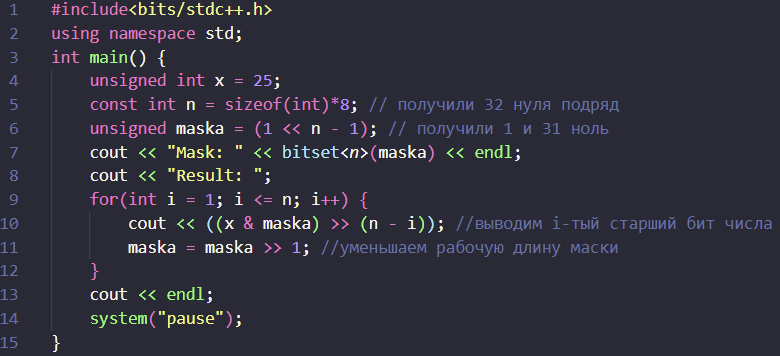
Ниже приведен код программ, решающий указанные выше задачи.



**Рисунок 1 – код программы к заданию 1.a**



**Рисунок 2 – код программы к заданию 1.б**



**Рисунок 3 – код программы к заданию 1.в**

* + 1. **Результаты тестирования**

Протестируем написанные программы, чтобы убедиться в их работоспособности.



**Рисунок 4 – результат работы программы к заданию 1.а**



**Рисунок 5 – результат работы программы к заданию 1.б**



**Рисунок 6 – результат работы программы к заданию 1.в**

Результаты тестирования соответствуют теоретическим вычислениям, следовательно код корректно решает поставленные задачи.

* 1. **Задание 2**
     1. **Формулировка задачи**

Используя сортировку при помощи битового массива, нужно последовательно решить три задачи:

* ввод произвольного набора до 8-ми чисел (со значениями [0, 7]) и его сортировка битовым массивом в виде числа unsigned char;
* адаптировать предыдущую программу для набора до 64-x чисел с битовым массивом типа unsigned long long;
* отсортировать массив, содержащий до 64-х значений, когда в качестве битового массива используется линейный массив чисел типа unsigned char.
  + 1. **Описание алгоритма**

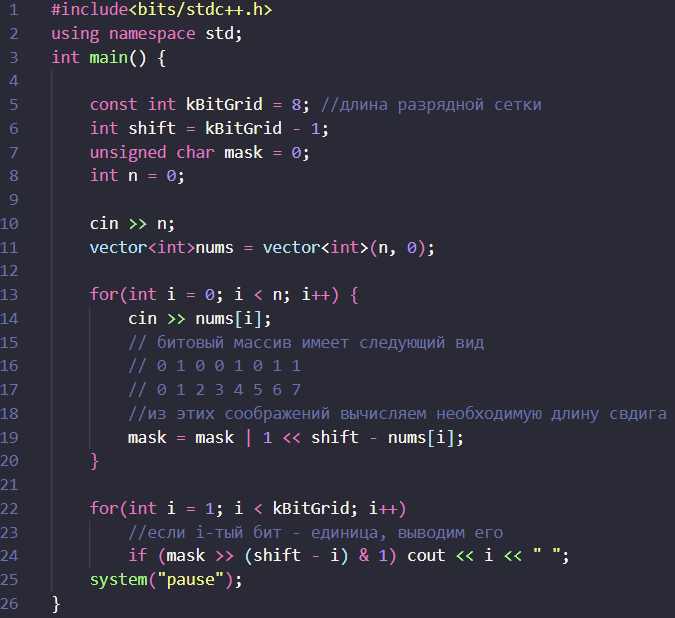
Для того, чтобы отсортировать массив, отобразим его в виде последовательности нулей и единиц, где 1 в i-том разряде – означает присутствие числа i в массиве. После этого последовательно считаем биты этой последовательности и выведем индексы тех разрядов, где стоит единица.

Как определять, с каким элементом битового массива надо взаимодействовать, когда в качестве него нужно будет линейный массив типа unsigned char? Целочисленное деление на количество бит в unsigned char (на 8) даст номер элемента битового массива, а результат от деления по модулю 8 укажет конкретный бит, который и будет представлять данное число.

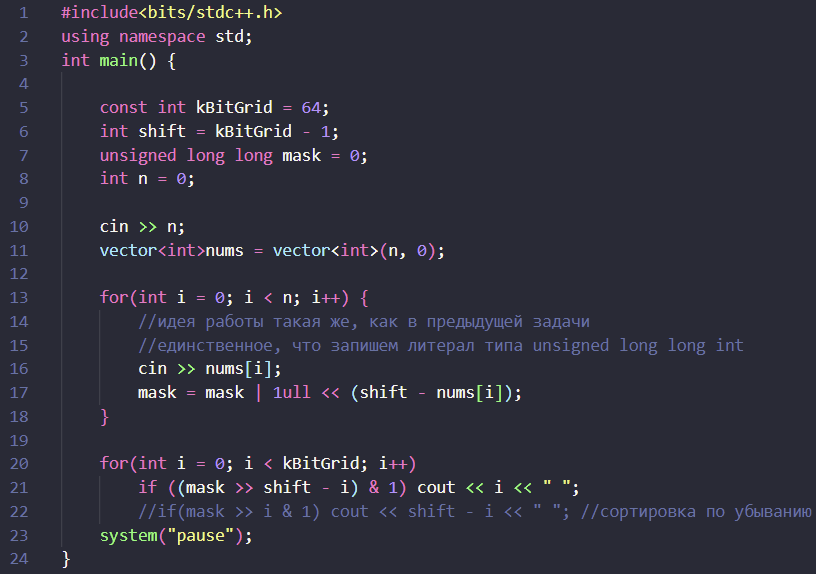
В целях оптимизации, можно добавлять новые блоки битового массива постепенно. Как только встречается число, которое нельзя записать в уже существующую разрядную сетку, добавим в цикле столько элементов типа unsigned char, сколько необходимо.

* + 1. **Код программы**

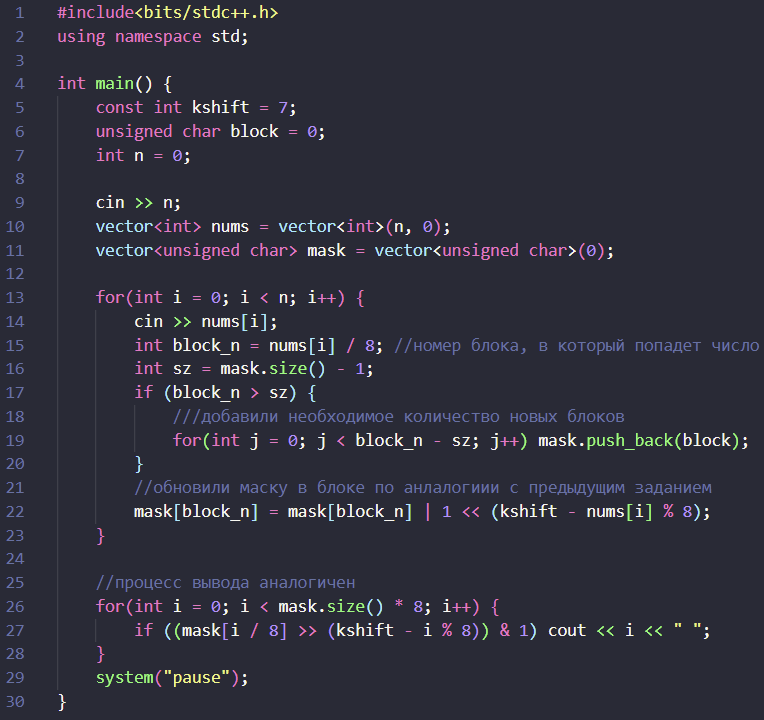
Реализуем описанный выше алгоритм.



**Рисунок 7 – код программы к заданию 2.а**



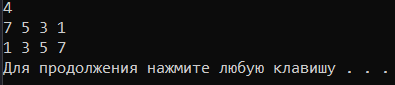
**Рисунок 8 – код программы к заданию 2.б**

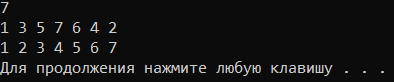


**Рисунок 9 – код программы к заданию 2.в**

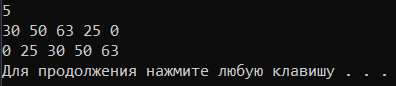
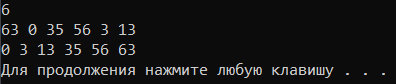
* + 1. **Результаты тестирования**

Протестируем написанные программы:

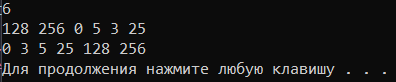


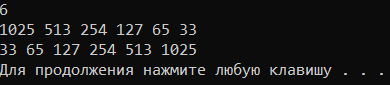


**Рисунки 10-11 – результаты работы программы к заданию 2.а**



**Рисунки 12-13 – результаты работы программы к заданию 2.б**





**Рисунки 14-15 – результаты работы программы к заданию 2.в**

Тестирование показало, что программы работают корректно, а код к заданию 2.в может сортировать любые значения, не привязываясь к разрядности сетки, при условии, что значения встречаются только один раз

* 1. **Задание 3**
     1. **Формулировка задачи**

Используя битовый массив реализовать высокоэффективную сортировку большого объема числовых данных в массиве. Необходимо учитывать следующие ограничения:

* время работы – 10с;
* максимальный объем используемой памяти – 1МБ.

Кроме того, требуется замерить время работы и объем занятой памяти.

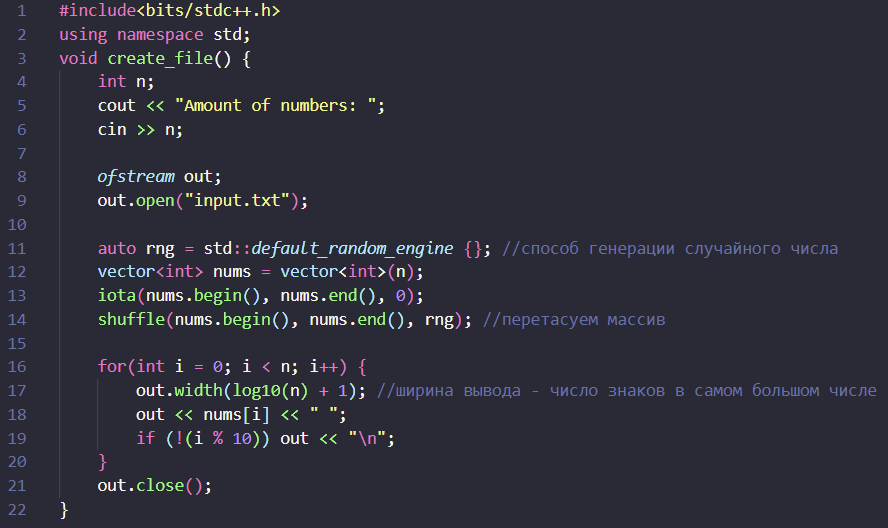
* + 1. **Описание алгоритма**

Генерация случайных уникальных чисел будет реализована так:

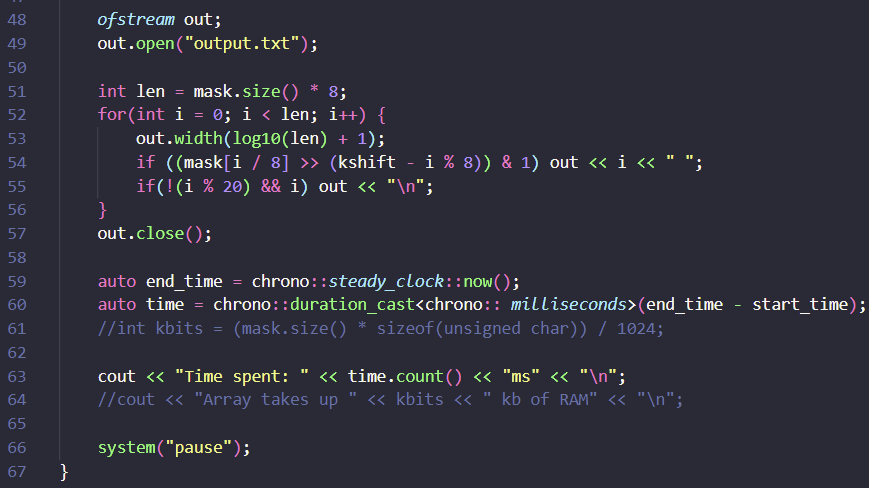
* создается vector, выбранного пользователем размера n
* при помощи функции iota он заполняется числами от 0 до n
* при помощи функции shuffle элементы массива перемешиваются случайным образом

Сам алгоритм сортировки работает аналогично тому, что использовался в задании 2.

* + 1. **Код программы**



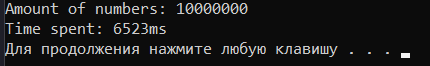


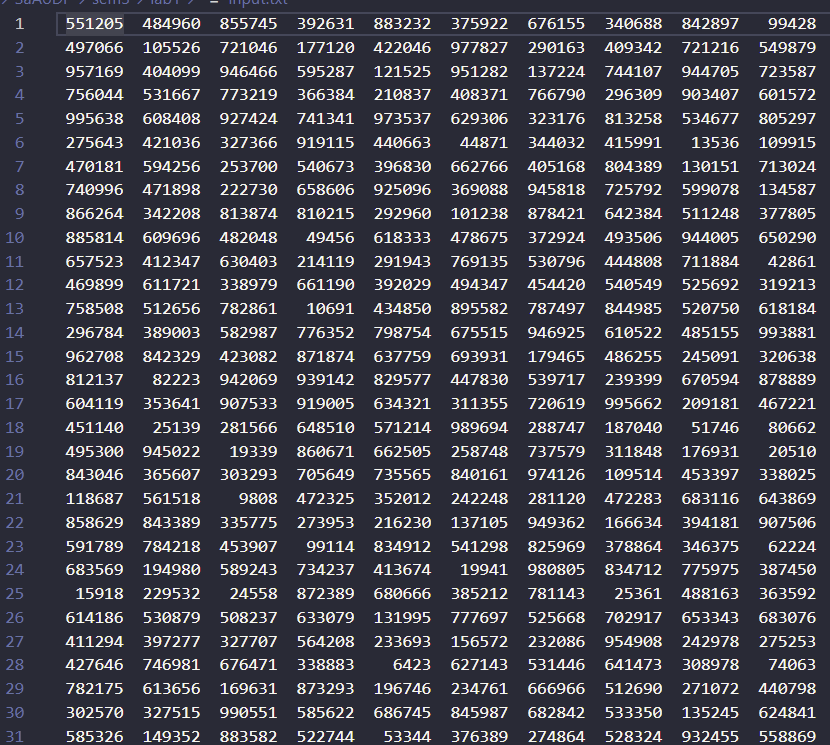


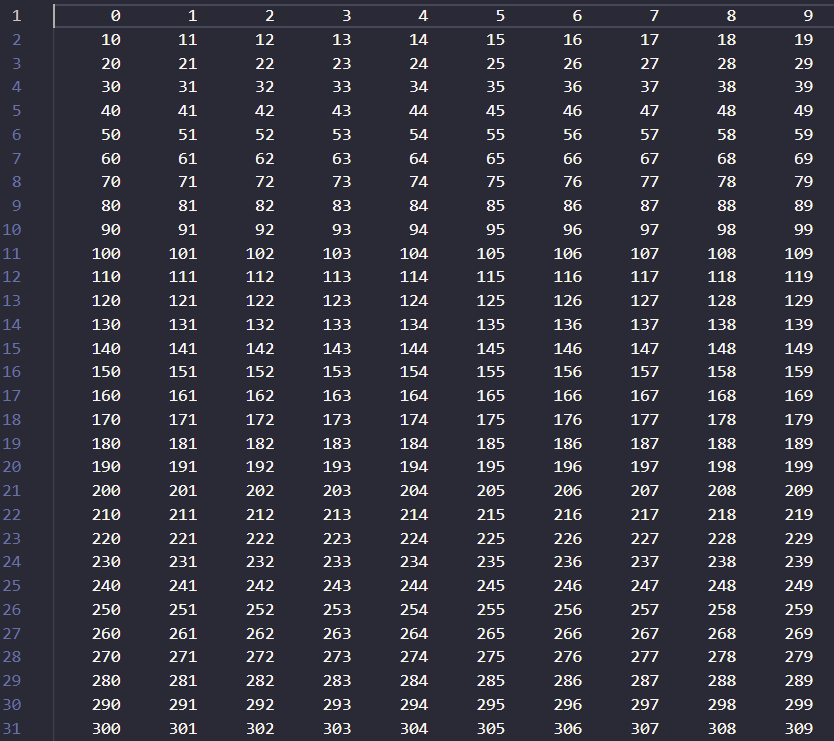
**Рисунки 16-18 – код программы к заданию 3.а и 3.б**

* + 1. **Результаты тестирования**

Протестируем программу на наибольшем количестве чисел – 107

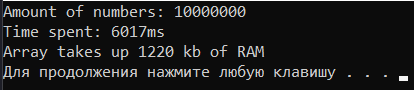






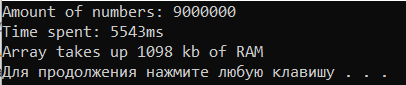
**Рисунки 19-21 – результаты работы программы к заданию 3.a(время работы, фрагмент входного файла, фрагмент выходного файла**)

Используя программные средства, выведем объем памяти, занимаемый при сортировке 107 чисел.



**Рисунок 22 – часть результата работы программы к заданию 3.б**

Получается, что программа может отсортировать чуть меньше 9×106 чисел, чтобы объем занимаемой памяти не превышал 1 МБ:



**Рисунок 23 – входные данные, удовлетворяющие ограничениям**

1. **Вывод**

В ходе работы были получены знания о приёмах работы с битовым представлением беззнаковых целых чисел, а также были реализованы несколько сортировок, в том числе эффективный алгоритм внешней сортировки на основе битового массива. Следовательно, можно говорить о достижении поставленной цели