****

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МИРЭА – Российский технологический университет»**

# РТУ МИРЭА

Отчет по выполнению практического задания 5.1

Тема: «РАБОТА С ДАННЫМИ ИЗ ФАЙЛА»

Дисциплина: Структуры и алгоритмы обработки данных

Выполнил студент Хамидуллин М.Р.

группа ИКБО-14-24

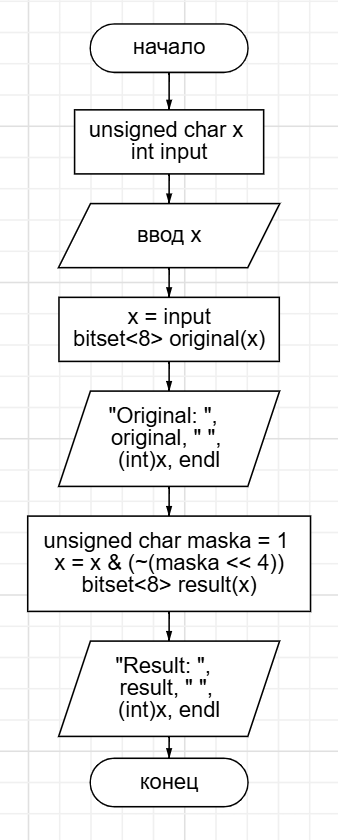
**Москва 2025**

**Цель работы:** освоить приёмы работы с битовым представлением беззнаковых целых чисел, реализовать эффективный алгоритм внешней сортировки на основе битового массива.

**Задание 1**

**1.a.) Реализуйте вышеприведённый пример, проверьте правильность результата в том числе и на других значениях х.**

**Математическая модель решения (описание алгоритма):**

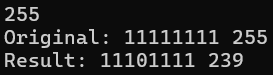
****

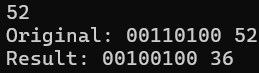
**Код программы:**

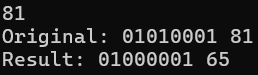
|  |
| --- |
| int a1\_a() {  unsigned char x;  int input;  cin >> input;  x = input;  bitset<8> original(x);  cout << "Original: " << original << " " << (int)x << endl;  unsigned char maska = 1;  x = x & (~(maska << 4));  bitset<8> result(x);  cout << "Result: " << result << " " << (int)x << endl;  cout << "===========================" << endl;  return 0;  } |

**Листинг 1. Задание 1.а.**

**Скрин-шоты тестирования:**

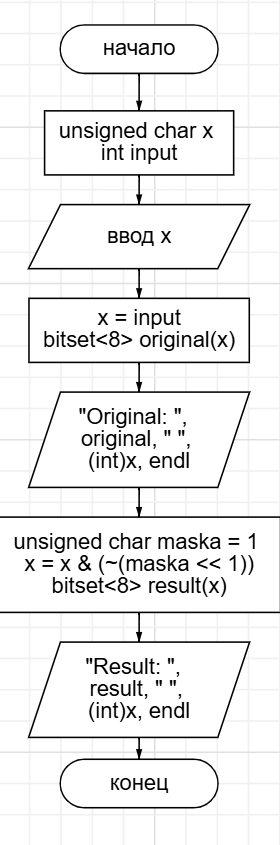
При x = 255: 

При x = 52: 

При x = 81: 

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

**Математическая модель решения (описание алгоритма):**

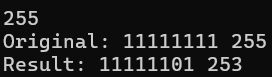


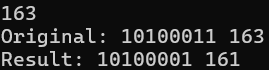
**Код программы:**

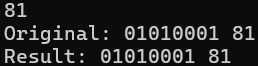
|  |
| --- |
| int a1\_b() {  unsigned char x;  int input;  cin >> input;  x = input;  bitset<8> original(x);  cout << "Original: " << original << " " << (int)x << endl;  unsigned char maska = 1;  x = x & (~(maska << 1));  bitset<8> result(x);  cout << "Result: " << result << " " << (int)x << endl;  cout << "===========================" << endl;  return 0;  } |

**Листинг 2. Задание 1.б.**

**Скрин-шоты тестирования:**

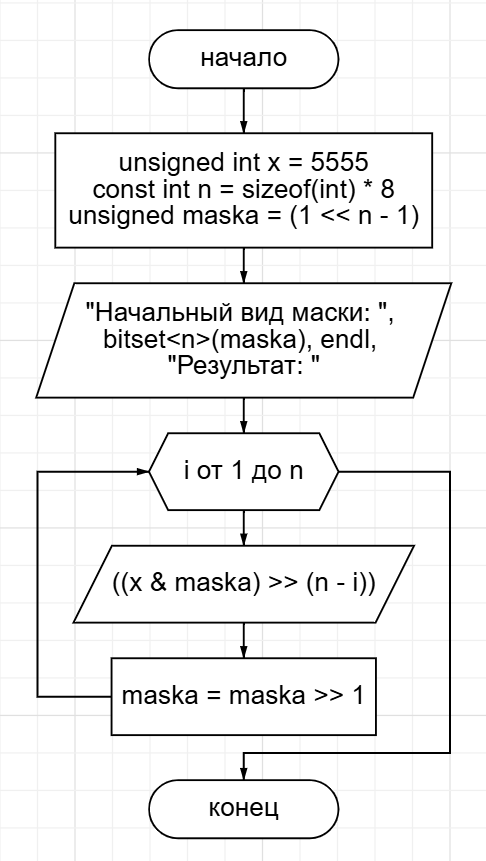
При x = 255: 

При x = 163: 

При x = 81: 

**1.в.) Реализуйте код листинга 1, объясните выводимый программой результат.**

**Математическая модель решения (описание алгоритма):**



**Код программы:**

|  |
| --- |
| void a1::a1\_c() {  SetConsoleCP(1251);  SetConsoleOutputCP(1251);  unsigned int x = 25;  const int n = sizeof(int) \* 8;  unsigned maska = (1 << n - 1);  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;  system("pause");  } |

**Листинг 3. Задание 1.в.**

**Скрин-шоты тестирования:**

При x = 25:



При x = 333:



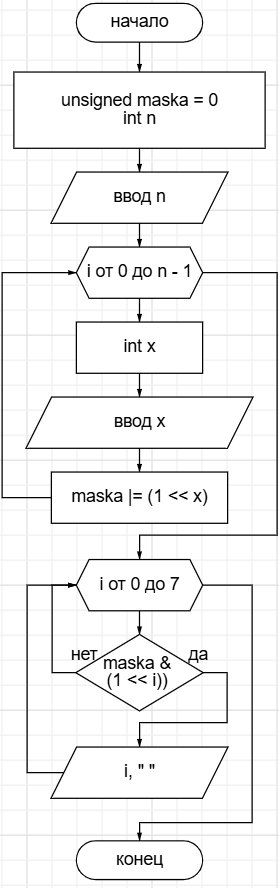
При x = 5555:

****

**Задание 2**

**2.a.) Реализуйте вышеописанный пример 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.**

**Математическая модель решения (описание алгоритма):**



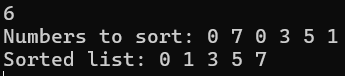
**Код программы:**

|  |
| --- |
| int a2\_a() {  unsigned maska = 0;  int n;  cin >> n;  cout << "Numbers to sort: ";  for (int i = 0; i < n; i++) {  int x;  cin >> x;  maska |= (1 << x);  }  cout << "Sorted list: ";  for (int i = 0; i < 8; i++) {  if (maska & (1 << i)) { cout << i << " "; }  }  cout << endl;  return 0;  } |

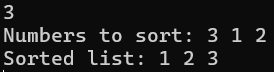
**Листинг 4. Задание 2.а.**

**Скрин-шоты тестирования:**

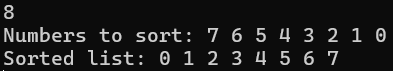
При n = 6 и x равных 0 7 0 3 5 1:



При n = 3 и x равных 3 1 2:

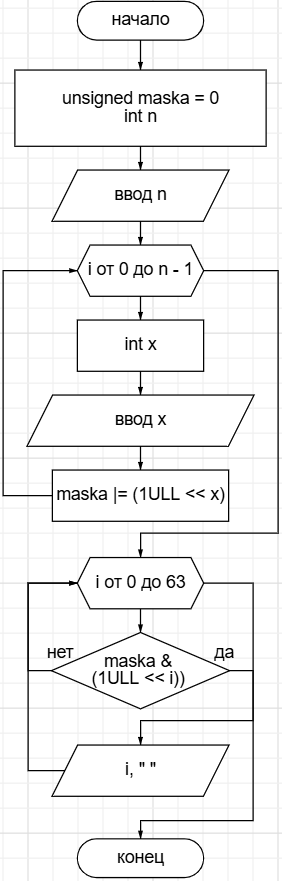


При n = 8 и x равных 7 6 5 4 3 2 1 0:



**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.**

**Математическая модель решения (описание алгоритма):**



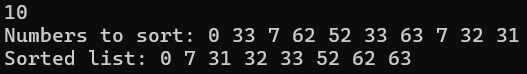
**Код программы:**

|  |
| --- |
| int a2\_b() {  unsigned long long maska = 0;  int n;  cin >> n;  cout << "Numbers to sort: ";  for (int i = 0; i < n; i++) {  int x;  cin >> x;  maska |= (1ULL << x);  }  cout << "Sorted list: ";  for (int i = 0; i < 64; i++) {  if (maska & (1ULL << i)) { cout << i << " "; }  }  cout << endl;  return 0;  } |

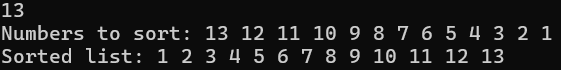
**Листинг 5. Задание 2.б.**

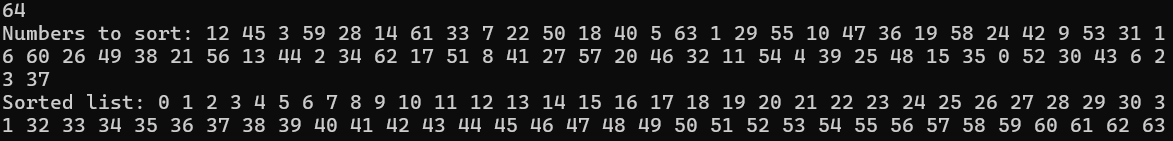
**Скрин-шоты тестирования:**

При n = 10 и x равных 0 33 7 62 52 33 63 7 32 31:



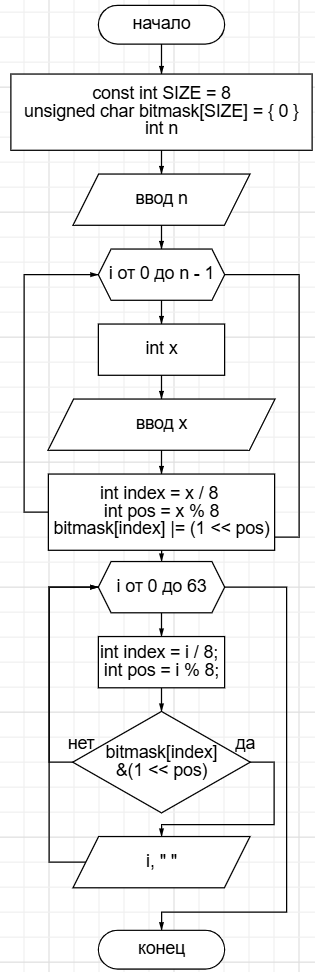
При n = 13 и x равных 13 12 11 10 9 8 7 6 5 4 3 2 1 :



При n = 64 и x равных 12 45 3 59 28 14 61 33 7 22 50 18 40 5 63 1 29 55 10 47 36 19 58 24 42 9 53 31 16 60 26 49 38 21 56 13 44 2 34 62 17 51 8 41 27 57 20 46 32 11 54 4 39 25 48 15 35 0 52 30 43 6 23 37:

**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):**



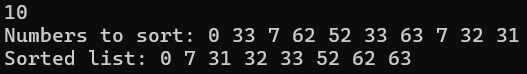
**Код программы:**

|  |
| --- |
| int a2\_c() {  const int SIZE = 8;  unsigned char bitmask[SIZE] = { 0 };  int n;  cin >> n;  cout << "Numbers to sort: ";  for (int i = 0; i < n; i++) {  int x;  cin >> x;  int index = x / 8;  int pos = x % 8;  bitmask[index] |= (1 << pos);  }  cout << "Sorted list: ";  for (int i = 0; i < 64; i++) {  int index = i / 8;  int pos = i % 8;  if (bitmask[index] & (1 << pos)) { cout << i << " "; }  }  cout << endl;  return 0;  } |

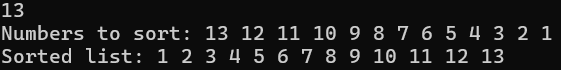
**Листинг 6. Задание 2.в.**

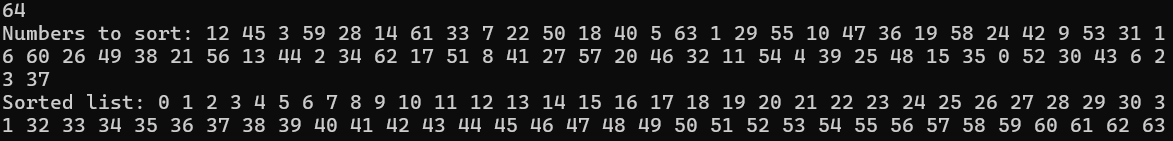
**Скрин-шоты тестирования:**

При n = 10 и x равных 0 33 7 62 52 33 63 7 32 31:



При n = 13 и x равных 13 12 11 10 9 8 7 6 5 4 3 2 1 :

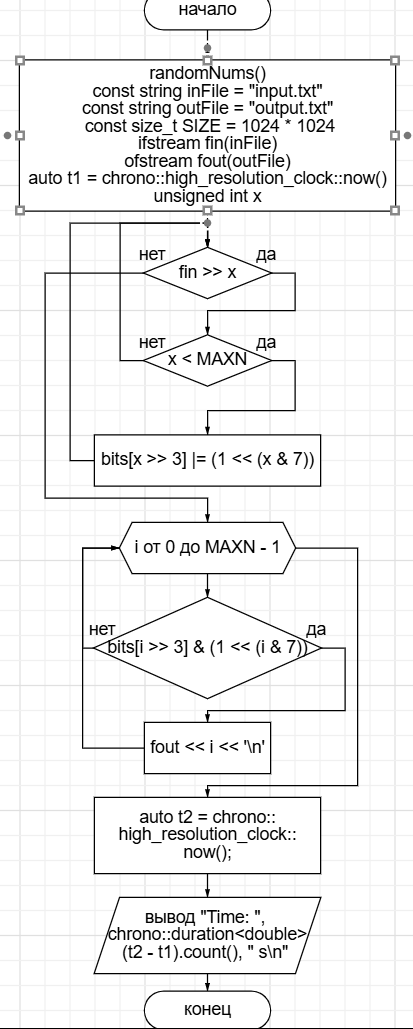


При n = 64 и x равных 12 45 3 59 28 14 61 33 7 22 50 18 40 5 63 1 29 55 10 47 36 19 58 24 42 9 53 31 16 60 26 49 38 21 56 13 44 2 34 62 17 51 8 41 27 57 20 46 32 11 54 4 39 25 48 15 35 0 52 30 43 6 23 37:

**Задание 3**

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

**Математическая модель решения (описание алгоритма):**

****

**Код программы:**

|  |
| --- |
| int a3\_a() {  randomNums();  const string inFile = "input.txt";  const string outFile = "output.txt";  const size\_t SIZE = 1024 \* 1024;  vector<uint8\_t> bits(SIZE, 0);  const size\_t MAXN = SIZE \* 8;  ifstream fin(inFile);  ofstream fout(outFile);  auto t1 = chrono::high\_resolution\_clock::now();  unsigned int x;  while (fin >> x) {  if (x < MAXN) bits[x >> 3] |= (1 << (x & 7));  }  for (size\_t i = 0; i < MAXN; i++)  if (bits[i >> 3] & (1 << (i & 7)))  fout << i << '\n';  auto t2 = chrono::high\_resolution\_clock::now();  cout << "Time: "  << chrono::duration<double>(t2 - t1).count()  << " s\n";  return 0;  } |

**Листинг 7. Задание 3.а.**

**Скрин-шоты тестирования:**



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

**Код программы:**

|  |
| --- |
| int a3\_a() {  randomNums();  const string inFile = "input.txt";  const string outFile = "output.txt";  const size\_t SIZE = 1024 \* 1024; //1 МБ  vector<uint8\_t> bits(SIZE, 0); //Битовый массив  const size\_t MAXN = SIZE \* 8; //Макс число, которое можно хранить  ifstream fin(inFile);  ofstream fout(outFile);  auto t1 = chrono::high\_resolution\_clock::now();  unsigned int x;  while (fin >> x) {  if (x < MAXN) bits[x >> 3] |= (1 << (x & 7));  }  for (size\_t i = 0; i < MAXN; i++)  if (bits[i >> 3] & (1 << (i & 7))) fout << i << '\n';  auto t2 = chrono::high\_resolution\_clock::now();  cout << "Time: "  << chrono::duration<double>(t2 - t1).count()  << " s\n";  cout << "Size massive in bit: " << bits.size() << endl;  return 0;  } |

**Листинг 8. Задание 3.б.**

**Скрин-шоты тестирования:**



**ЛИТЕРАТУРА**

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).