

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 ПРАКТИЧЕСКОЙ РАБОТЕ №1**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

Тема. Поразрядные операции и их применение

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы ИКБО-04-22 |  | Кликушин В.И. |
| Принял старший преподаватель |  | Скворцова Л.А. |

Москва 2023

ОГЛАВЛЕНИЕ

[1 ОТЧЁТ ПО ЗАДАНИЮ 1 4](#_Toc145107525)

[1.1 ПОСТАНОВКА ЗАДАЧИ 4](#_Toc145107526)

[1.2 УПРАЖНЕНИЕ 1 4](#_Toc145107527)

[1.2.1 Номер и условие упражнения 4](#_Toc145107528)

[1.2.2 Разработка решения задачи 4](#_Toc145107529)

[1.2.3 Реализация решения 4](#_Toc145107530)

[1.2.4 Результат тестирования 5](#_Toc145107531)

[1.3 УПРАЖНЕНИЕ 2 5](#_Toc145107532)

[1.3.1 Номер и условие упражнения 5](#_Toc145107533)

[1.3.2 Разработка решения задачи 5](#_Toc145107534)

[1.3.3 Реализация решения 5](#_Toc145107535)

[1.3.4 Результат тестирования 6](#_Toc145107536)

[1.4 УПРАЖНЕНИЕ 3 6](#_Toc145107537)

[1.4.1 Номер и условие упражнения 6](#_Toc145107538)

[1.4.2 Разработка решения задачи 6](#_Toc145107539)

[1.4.3 Реализация решения 6](#_Toc145107540)

[1.4.4 Результат тестирования 7](#_Toc145107541)

[1.5 УПРАЖНЕНИЕ 4 7](#_Toc145107542)

[1.5.1 Номер и условие упражнения 7](#_Toc145107543)

[1.5.2 Разработка решения задачи 7](#_Toc145107544)

[1.5.3 Реализация решения 7](#_Toc145107545)

[1.5.4 Результат тестирования 8](#_Toc145107546)

[1.6 УПРАЖНЕНИЕ 5 8](#_Toc145107547)

[1.6.1 Номер и условие упражнения 8](#_Toc145107548)

[1.6.2 Разработка решения задачи 8](#_Toc145107549)

[1.6.3 Реализация решения 8](#_Toc145107550)

[1.6.4 Результат тестирования 9](#_Toc145107551)

[2 ОТЧЁТ ПО ЗАДАНИЮ 2 10](#_Toc145107552)

[2.1 ПОСТАНОВКА ЗАДАЧИ 10](#_Toc145107553)

[2.2 ОПИСАНИЕ СТРУКТУРЫ, ИСПОЛЬЗУЕМОЙ В ЗАДАЧЕ 10](#_Toc145107554)

[2.3 АЛГОРИТМ РЕШЕНИЯ 10](#_Toc145107555)

[2.4 ТЕСТОВЫЙ ПРИМЕР 10](#_Toc145107556)

[2.5 КОД ПРОГРАММЫ 10](#_Toc145107557)

[2.5 СКРИНЫ РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ 11](#_Toc145107558)

[3 ВЫВОДЫ 13](#_Toc145107559)

[4 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ 14](#_Toc145107560)

# 1 ОТЧЁТ ПО ЗАДАНИЮ 1

## 1.1 ПОСТАНОВКА ЗАДАЧИ

Выполнить упражнения по применению битовых операций по изменению значений битов в ячейке оперативной памяти, созданию маски для изменения значения ячейки. Создание выражения, содержащего поразрядные операции, для выполнения определенной операции над значением ячейки. Разработать программу, которая продемонстрирует выполнение упражнений варианта. Результаты выполнения упражнения выводить на монитор. Задания в соответствие с вариантом №13: установить 7,9,11 биты в единицу; обнулить биты с четными номерами; умножить число на 128, используя поразрядную операцию, разделить число на 128, используя поразрядную операцию, обнулить n-ый бит.

## 1.2 УПРАЖНЕНИЕ 1

### 1.2.1 Номер и условие упражнения

Определить переменную целого типа, присвоить ей значение, используя константу в шестнадцатеричной системе счисления. Разработать оператор присваивания и его выражение, которое установит 7,9,11 биты исходного значения переменной в значение 1, используя соответствующую маску и поразрядную операцию.

### 1.2.2 Разработка решения задачи

Выражение: x = x | (1 << 7 | 1 << 9 | 1 << 11);

Маска: maska = 0xF57F;

Пример: x = 0000 0000 0000 0000

Результат: 0000 1010 1000 0000

### 1.2.3 Реализация решения

unsigned short task\_1(unsigned int x)

{

return x | (1 << 7 | 1 << 9 | 1 << 11);

}

### 1.2.4 Результат тестирования

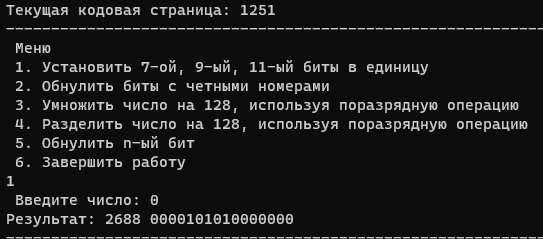


Рисунок 1 – Результат теста 1.1

## 1.3 УПРАЖНЕНИЕ 2

### 1.3.1 Номер и условие упражнения

Определить переменную целого типа. Разработать оператор присваивания и его выражение, которое обнуляет чётные биты исходного значения переменной, используя соответствующую маску и поразрядную операцию. Значение в переменную вводится с клавиатуры.

### 1.3.2 Разработка решения задачи

Маска: maska = 0xAAAA;

Пример: x = 1111 1011 0010 0101

Результат: 1010 1010 0010 0000

### 1.3.3 Реализация решения

unsigned short task\_2(unsigned int x)

{

unsigned short int maska = 0xAAAA;

return x & maska;

}

### 1.3.4 Результат тестирования

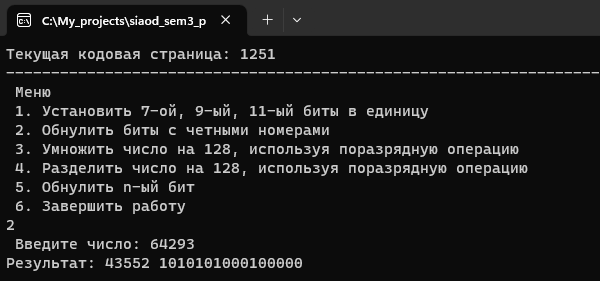


Рисунок 2 – Результат теста 1.2

## 1.4 УПРАЖНЕНИЕ 3

### 1.4.1 Номер и условие упражнения

Определить переменную целого типа. Разработать оператор присваивания и выражение, которое умножает значение переменной на 128, используя соответствующую поразрядную операцию. Изменяемое число вводится с клавиатуры.

### 1.4.2 Разработка решения задачи

Выражение: x = x << 7;

Пример: x = 0000 0000 0001 1001

Результат: 0000 1100 1000 0000

### 1.4.3 Реализация решения

unsigned short task\_3(unsigned int x)

{

return x << 7;

}

### 1.4.4 Результат тестирования

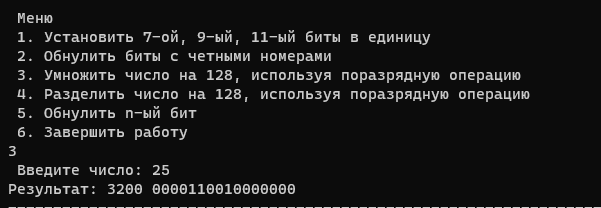


Рисунок 3 – Результат теста 1.3

## 1.5 УПРАЖНЕНИЕ 4

### 1.5.1 Номер и условие упражнения

Определить переменную целого типа. Разработать оператор присваивания и выражение, которое делит значение переменной на 128, используя соответствующую поразрядную операцию. Изменяемое число вводится с клавиатуры.

### 1.5.2 Разработка решения задачи

Выражение: x = x >> 7;

Пример: x = 0000 1100 1000 0000

Результат: 0000 0000 0001 1001

### 1.5.3 Реализация решения

unsigned short task\_4(unsigned int x)

{

return x >> 7;

}

### 1.5.4 Результат тестирования

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 – Результат теста 1.4

## 1.6 УПРАЖНЕНИЕ 5

### 1.6.1 Номер и условие упражнения

Определить переменную целого типа. Разработать оператор присваивания и выражение, в котором используются только поразрядные операции. В выражении используется маска – переменная. Маска может быть инициализирована единицей в младшем разряде (вар 1) или единицей в старшем разряде (вар 2). Изменяемое число вводится с клавиатуры. Обнулить n-ый бит, используя маску 1.

### 1.6.2 Разработка решения задачи

Выражение: x = x & (~(maska << num));

Пример: x = 0100 1111 1000 0000

Результат: 0000 1111 1000 0000

### 1.6.3 Реализация решения

unsigned short task\_5(unsigned int x, int num)

{

unsigned short int maska = 1;

return x & (~(maska << num));

}

### 1.6.4 Результат тестирования

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 – Результат теста 1.5

# 2 ОТЧЁТ ПО ЗАДАНИЮ 2

## 2.1 ПОСТАНОВКА ЗАДАЧИ

Разработать программу сортировки большого объема данных, хранящихся в файле. Дано: файл, содержащий не более n (n=107) целых положительных чисел, каждое из которых семизначное число, т.е. принадлежит диапазону [1000000..9999999] и среди них нет повторяющихся. Результат: упорядоченный по возрастанию список чисел, сохраненных в файле. Ограничения: доступной оперативной памяти 1МБ, дисковая память неограниченна, время выполнения программы 10 секунд.

## 2.2 ОПИСАНИЕ СТРУКТУРЫ, ИСПОЛЬЗУЕМОЙ В ЗАДАЧЕ

В качестве структуры для хранения данных был выбран динамический массив типа unsigned char. Чтобы уместить 107 чисел необходимо задать массив размерности (107/8)+1 элементов, так как размерность char – один байт.

## 2.3 АЛГОРИТМ РЕШЕНИЯ

1. Необходимо объявить и инициализировать массив битов нулями
2. Заполнить массив значениями, вводимыми с клавиатуры (из файла)
3. Перебрать каждый элемент массива «побитово», если бит будет установлен в единицу, то номер этого бита – число в отсортированном массиве.

## 2.4 ТЕСТОВЫЙ ПРИМЕР

В качестве тестового примера рассмотрим следующую последовательность для сортировки: 1, 10, 4, 7, 5, 14, 99, 777.

Индекс, на котором будет расположен элемент n можно посчитать как n/8 или n>>3. А номер бита в числе, хранимом в этой ячейки – остаток от деления n%8 или же n&7. Таким образом, единица будет помещена в ячейку с нулевым индексом, двоичное представление числа, хранящегося в этой ячейке: 00..0010 и тд.

## 2.5 КОД ПРОГРАММЫ

#include <iostream>

#include <chrono>

using namespace std;

int main()

{

auto start = chrono::high\_resolution\_clock::now();

const int N = pow(10,7);

system("chcp 1251");

int size=1000,n;

//cout << " Введите количество элементов в массиве: ";

//cin >> size;

unsigned char\* arr = new unsigned char[N];

for (int i = 0; i < (N>>3)+1; i++)

{

arr[i] = 0;

}

unsigned maska = 1;

srand(time(NULL));

for (int i = 0; i < size; i++)

{

//cin >> n;

n = rand() % 1000;

arr[n >> 3] = arr[n >> 3] | (maska << (n & 7));

}

for (int i = 0; i < (N>>3)+1; i++)

{

for (int j = 0; j < 8; j++)

{

n = (arr[i] & maska) >> j;

if (n == 1)

cout << (i<<3) + j << " ";

maska <<= 1;

}

maska = 1;

}

auto end = chrono::high\_resolution\_clock::now();

auto len = chrono::duration\_cast<chrono::nanoseconds>(end - start);

cout << "Время выполнения: " << len.count() / 1e6 << " мс\n";

delete[] arr;

}

## 2.5 СКРИНЫ РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ

Время выполнения указано в консоле.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – Результат теста из примера

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Рисунок 7 – Результат теста для входных данных объема 100 чисел

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 8 – Результат теста для входных данных объема 1000 чисел

# 3 ВЫВОДЫ

В ходе выполнения практической работы мной были получены навыки работы с побитовыми операциями, а также рассмотрен нестандартный подход к сортировке с использованием последовательности бит для представления чисел.

# 4 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ

1. Учебно-методическое пособие СиАОД (часть 2)

2. Приложение к практическим работам – СДО (online-edu.mirea.ru)