МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий (ИТ)

Кафедра математического обеспечения и стандартизации информационных

технологий (МОСИТ)

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1**

**по дисциплине «***Структуры и алгоритмы обработки данных***»**

**тема «**Битовые операции над структурами данных**»**

Выполнил: студент группы ИВБО-02-15 Прохоров А.В.

Приняла: старший преподаватель кафедры Скворцова Л.А.

**1. Условие задания**

**Цель.**

* Получить знания по операциям языка программирования С++ для обработки битовой информации.
* Получение практических навыков по выполнению операций над битовым представлением данных

**Задание.**

1. Разработать программу, которая реализует задачу варианта, используя в операциях внутреннее представление данных – их двоичный код.
2. Требования.

Выполнить декомпозицию задачи

Выделенные подзадачи оформить в виде подпрограмм.

**Вариант №20:**

Алгоритм шифрования перемешиванием:

Дано число, переставьте его соседние биты (то есть поменяйте местами биты с номерами 0 и 1, 2 и 3, 4 и 5 и т.д.). Разрешается использовать битовые операции. Запрещается использовать арифметические операции, ветвления, циклы. Общее число бит в числе не превосходит 32. (Например, для числа 78 результат 141).

**2. Определение структуры представления данных и операций над данными**

**Бит** (от англ. ***bi****nary digi****t*** — двоичное число) — единица измерения количества информации.

**Операции над разрядами:**

1. **Сдвиг влево** двоичного кода (умножение на 2)

Формат:

x<<n

где х - значение, код которого должен быть изменен

n – Количество разрядов, на который должен быть сдвинут код

2. **Сдвиг вправо** двоичного кода (деление на 2)

Формат:

x>>n

где х - значение, код которого должен быть изменен, n – Количество разрядов, на который должен быть сдвинут код

3. **Поразрядное сложение**

Используется для установки в 1 заданного бита

Формат:

X|Y

X и Y переменные одной разрядности

Таблица битового сложения

0|1=1

0|0=0

1|0=1

1|1=1

4. **Поразрядное умножение**

Используется для обнуления заданного бита

Формат:

X & Y

Таблица битового сложения

0&1=0

0&0=0

1&0=0

1&1=1

5. **Инвертирование**

Замена бита 0 на значение 1 и наоборот.

**3. Разработка программы**

**3.1. Декомпозиция**

Список подзадач:

1) Создание масок введенного числа для переноса пар

2) Условие переноса и его отсутствия

3) Вывод обработанного числа

**3.2. Определение функций**

unsigned int swap\_bits — смена позиции заданных битов. Получает n — число, которое требуется перевести, а также i и j — позиции битов, которые нужно поменять местами.

**3.3. Код разработанного приложения**

#include <iostream>

#include <cstdlib>

using namespace std;

unsigned int swap\_bits(unsigned int n, int i, int j){

unsigned int a, b;

a = (n >> i) & 1;

b = (n >> j) & 1;

if(a != b)

n = n & ~((1 << i) | (1 << j)) | (a << j) | (b << i);

return n;

}

int main(){

setlocale(LC\_ALL, "Russian");

unsigned int n;

cout << "Введите исходное число: ";

cin >> n;

n = swap\_bits(n, 0, 1);

n = swap\_bits(n, 2, 3);

n = swap\_bits(n, 4, 5);

n = swap\_bits(n, 6, 7);

n = swap\_bits(n, 8, 9);

n = swap\_bits(n, 10, 11);

n = swap\_bits(n, 12, 13);

n = swap\_bits(n, 14, 15);

n = swap\_bits(n, 16, 17);

n = swap\_bits(n, 18, 19);

n = swap\_bits(n, 20, 21);

n = swap\_bits(n, 22, 23);

n = swap\_bits(n, 24, 25);

n = swap\_bits(n, 26, 27);

n = swap\_bits(n, 28, 29);

n = swap\_bits(n, 30, 31);

cout << "Конечное число: " << n <<endl;

system("Pause");

return 0;

}

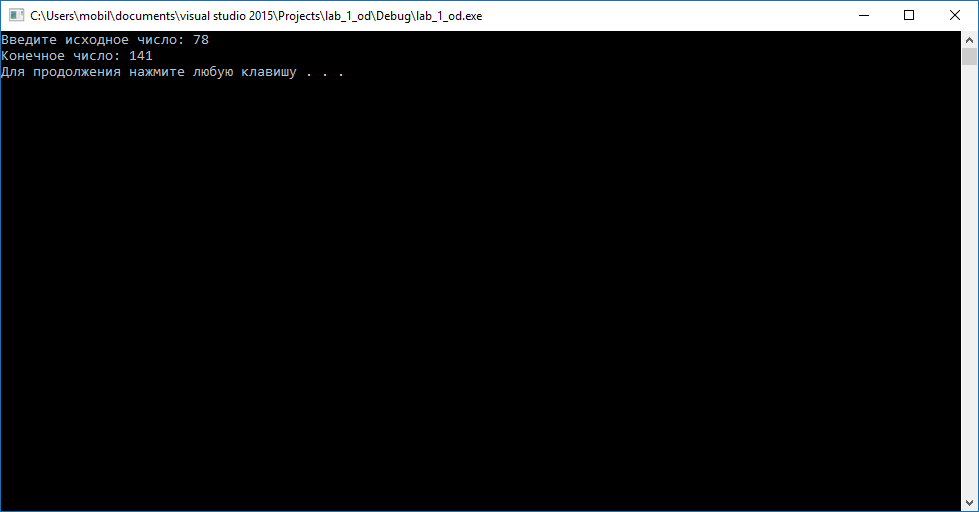
**4. Тестирование программы**

**4.1. Таблица тестов**

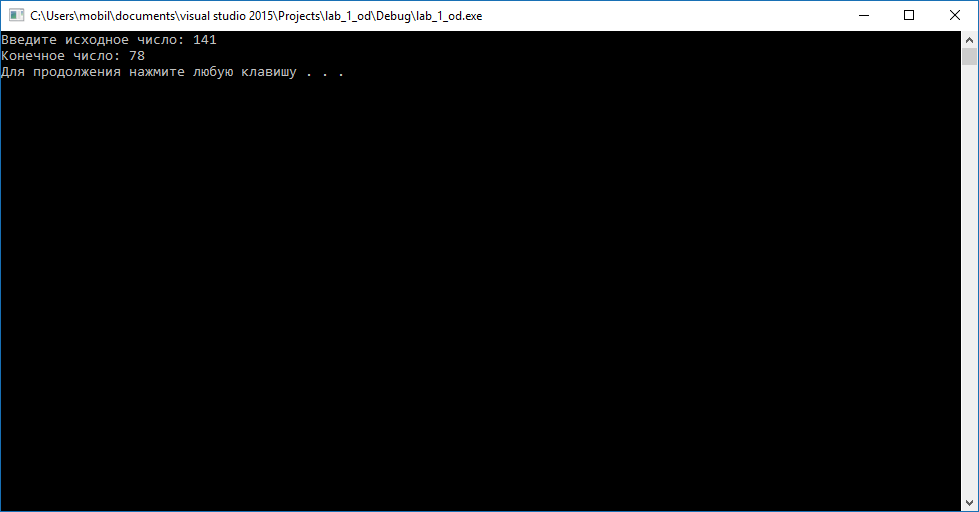
|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Введенное число | Битовое представление | Результат работы |
| 1 | 78 | 01001110 | 141 |
| 2 | 141 | 10001101 | 78 |
| 3 | 25 | 011001 | 38 |
| 4 | 38 | 100110 | 25 |

**4.2. Контрольные прогоны программы**

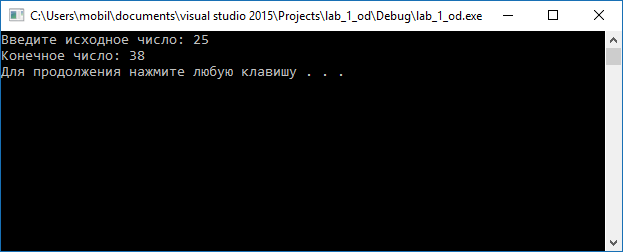
Тест №1 — ввод числа, указанного в задании (78).



Тест №2 — ввод обратного значения, позволяющего сделать проверку алгоритма шифрования.



Тест №3 — ввод числа 25.



Тест №4 — обратная проверка теста №3 — ввод числа 38.

