|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1** | |
| **по дисциплине** | |
| **«**Структуры и алгоритмы обработки данных**»**  **Тема: «Поразрядные операции и их применение»** | |
|  | |
| Выполнил студент группы ИКБО-06-21 | Хоров Н.М. |
| Принял преподаватель | Филатов А.С. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2022

# **Цель работы**

Получение навыков применения поразрядных операций в алгоритмах.

# **Постановка задачи**

1. Разработать программу, которая продемонстрирует выполнение упражнений варианта. Результаты выполнения упражнения выводить на монитор.

Требования к упражнениям:

1. Определить переменную целого типа, присвоить ей значение, используя константу в шестнадцатеричной системе счисления. Разработать функцию, которое установит заданные в задании биты исходного значения переменной в значение 1, используя соответствующую маску и поразрядную операцию.
2. Разработать функцию, которая обнуляет заданные в задании биты исходного значения целочисленной переменной, введенной пользователем, используя соответствующую маску и поразрядную операцию.
3. Разработать функцию, которая умножает значение целочисленной переменной, введенной пользователем, на множитель, используя соответствующую поразрядную операцию.
4. Разработать функцию, которая делит значение целочисленной переменной, введенной пользователем, на делитель, используя соответствующую поразрядную операцию.
5. Разработать функцию, реализующую задание, в которой используются только поразрядные операции. В выражении используется маска – переменная. Маска инициализируется единицей в младшем разряде (маска 1) или единицей в старшем разряде (маска 2). Изменяемое число и n вводится с клавиатуры.
6. Провести тестирование программы на небольших объемах данных, введенных вручную. Разработанные тесты должны покрывать все случаи входных данных (средний, лучший, худший). Результаты тестирования свести в сводные таблицы.
7. Составить отчет, отобразив в нем описание выполнения всех этапов разработки, тестирования и код всей программы со скриншотами результатов тестирования.

Вариант №1. Условие задания:

|  |  |
| --- | --- |
| Упражнение 1 | 3-ий, 11-ый, 5-ый |
| Упражнение 2 | Четыре младших |
| Упражнение 3 | 128 |
| Упражнение 4 | 128 |
| Упражнение 5 | Установить n-ый бит в 1, используя маску 2 |

# **Решение**

В С++ существуют шесть поразрядных логических операций:

* поразрядное И, обозначение: &
* поразрядное исключающее ИЛИ, обозначение:  ^
* поразрядное включающее ИЛИ, обозначение:  |
* поразрядное НЕ, обозначение: ~
* поразрядное умножение, побитовый сдвиг вправо, обозначение: >>
* поразрядное деление, побитовый сдвиг влево, обозначение: <<

Данные операции работают с битами ячеек памяти, причём операнды и результат могут быть заданы в другой форме, например, в десятичной.

Функция convert\_to\_bin принимает целочисленную переменную num. Далее:

* Создается пустая строковая переменная result.
* Проверяется, не нулевое ли значение num. Если num равно 0, то возвращаемое значение будет выглядеть как “0”.
* После создается цикл while с условием “num>0”.
* В строку поразрядно добавляется новый бит, начиная с конца, с помощью функции to\_string. После num нацело делится на 2. Далее возвращается результат.

|  |
| --- |
| //Функция перевода чисел с десятичной системы счисления в двоичную систему счисления с помощью поразрядных операций  string convert\_to\_bin(unsigned int num)  {  unsigned int t = 0x80000000;  string result = "";  if (num == 0) return "0";  while (t>0) {    if (t & num) {  result += "1";  }  else {  result += "0";  }  t /= 2;  }  return result;    } |
|  |

* 1. **Решение первого задания.**

В первом задании необходимо определить переменную целого типа, присвоить ей значение, используя константу в шестнадцатеричной системе счисления. Разработать функцию, которое установит заданные в задании биты (в нашем случае 3-ий, 11-ый, 5-ый) исходного значения переменной в значение 1, используя соответствующую маску и поразрядную операцию.

Для решения поставленной задачи установим маске значение 2088 в 10СС, которое соответствует единицам в 3-ем, 5-ом и 11-ом битах в 2СС. При выводе чисел в консоль, используем поразрядную операцию «ИЛИ», а также функцию для перевода чисел в двоичную систему счисления.

Тестирование:

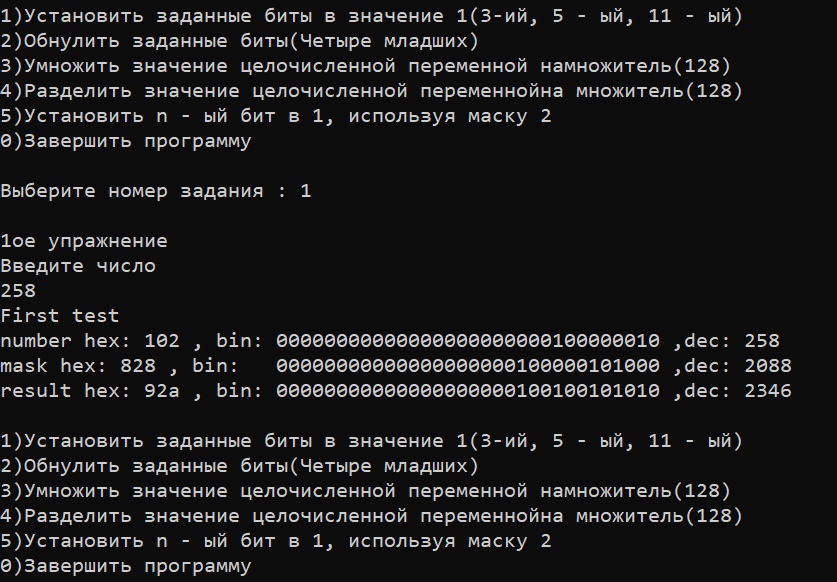


Рисунок 1 - тест 1

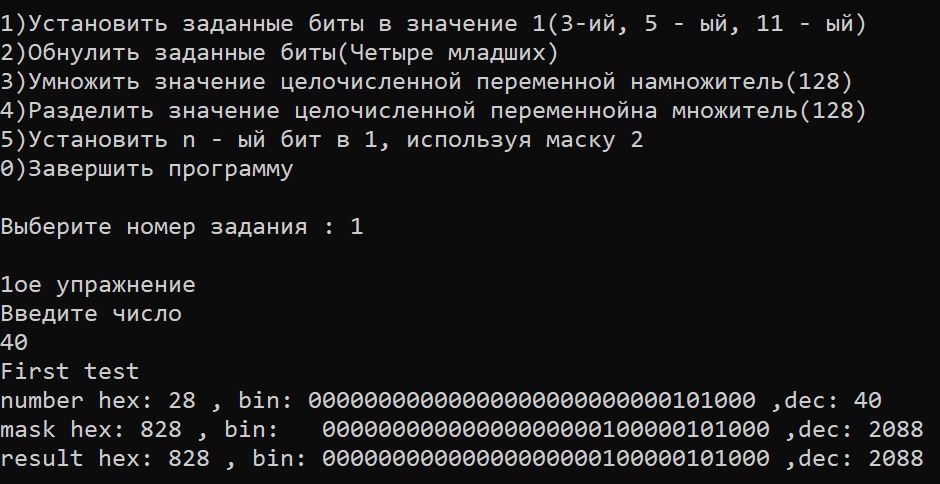


Рисунок 2 - тест 2

**3.2 Решение второго задания**

Во втором задании необходимо разработать функцию, которая обнуляет заданные в задании биты (в нашем случае четыре младших) исходного значения целочисленной переменной, введенной пользователем, используя соответствующую маску и поразрядную операцию.

Для решения поставленной задачи установим маске значение FFFFFFF0 в 16СС, которое соответствует 28 единицам и 4 нулям в 2СС. При выводе чисел в консоль, используем поразрядную операцию «И», а также функцию для перевода чисел в двоичную систему счисления.

Тестирование:

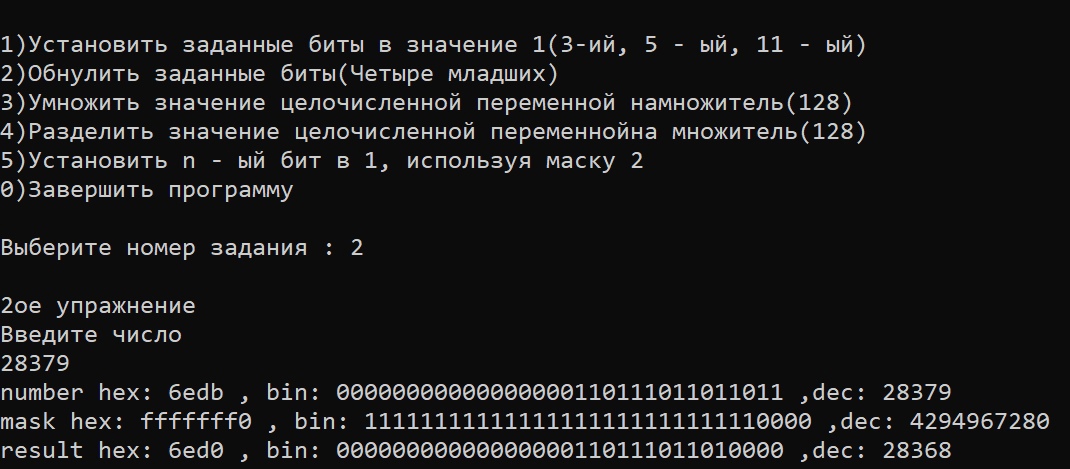


Рисунок 3

* 1. **Решение третьего задания**

В третьем задании необходимо разработать функцию, которая умножает значение целочисленной переменной, введенной пользователем, на множитель (128), используя соответствующую поразрядную операцию.

Для решения поставленной задачи установим переменной digit значение 7 в соответствии с данным в задании числом 128=2^7. Используем функцию third, которая отвечает за побитовый сдвиг влево, а также функцию convert\_to\_int при выводе в консоль.

|  |
| --- |
| //Функция для побитового умножения числа  void third(unsigned int& numb, int& digit) {    numb = numb << digit;  } |
|  |

Тестирование:

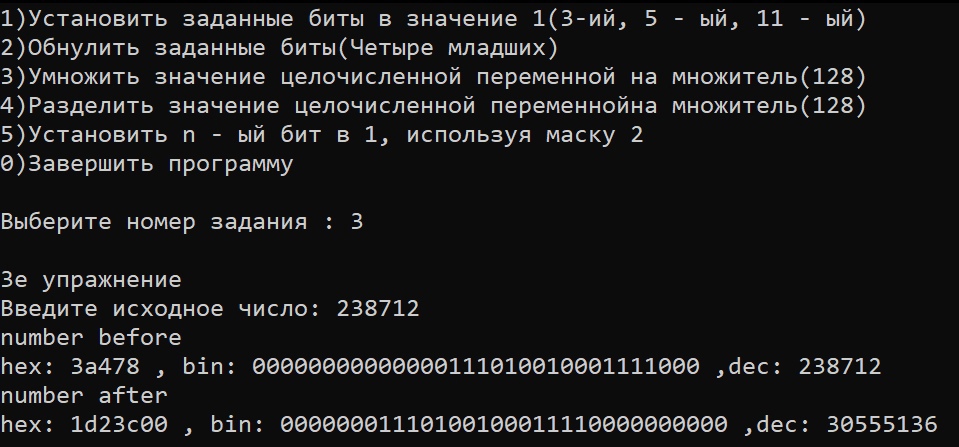


Рисунок 4

* 1. **Решение четвертого задания**

В четвертом задании необходимо разработать функцию, которая делит значение целочисленной переменной, введенной пользователем, на множитель (128), используя соответствующую поразрядную операцию.

Для решения поставленной задачи установим переменной digit значение 7 в соответствии с данным в задании числом 128=2^7. Используем функцию fourth, которая отвечает за побитовый сдвиг вправо, а также функцию convert\_to\_int при выводе в консоль.

|  |
| --- |
| //Функция для побитового деления числа  void fourth(unsigned int& numb, int& digit) {    numb = numb >> digit;    } |
|  |

Тестирование:

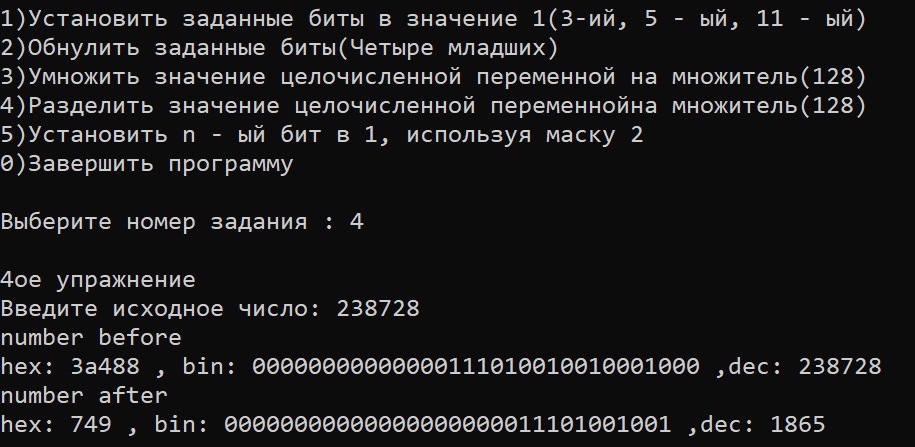


Рисунок 5

* 1. **Решение четвертого задания**

В пятом задании необходимо разработать функцию, реализующую задание, в которой используются только поразрядные операции. В выражении используется маска – переменная. Маска инициализируется единицей в младшем разряде (маска 1) или единицей в старшем разряде (маска 2), в нашем случае – единицей в старшем разряде. Изменяемое число и n вводится с клавиатуры.

Для решения поставленной задачи используем маску равную 80000000 в 16СС, соответствующую единице и 31-ому нулю в двоичной СС. Функция fifth отвечает за добавление в соответствующий с введенной переменной n разряд единицу. Используем функцию convert\_to\_int при выводе в консоль.

|  |
| --- |
| //Функция для добавление в разряд n единицы  void fifth(unsigned int& numb, unsigned int& mask, int& n) {  mask = mask >> n;  numb = numb | mask;  } |
|  |

Тестирование:

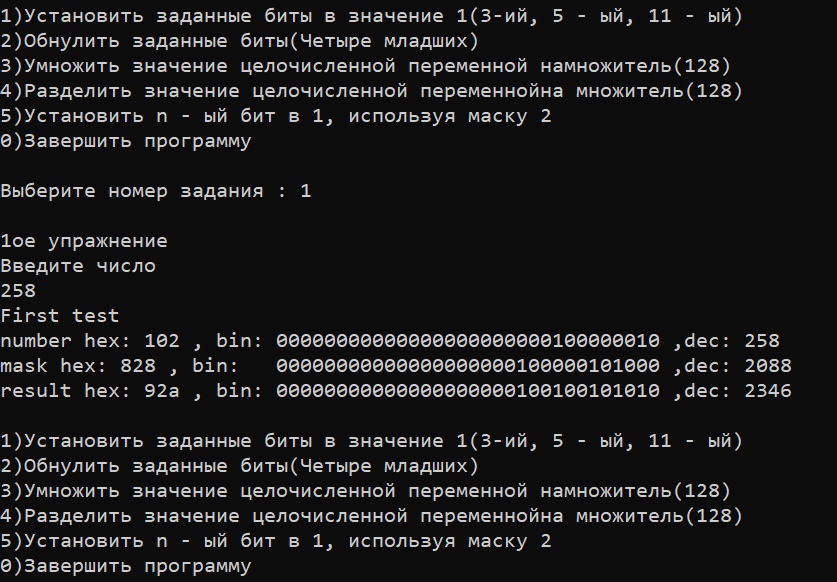


Рисунок 6 - тест 1

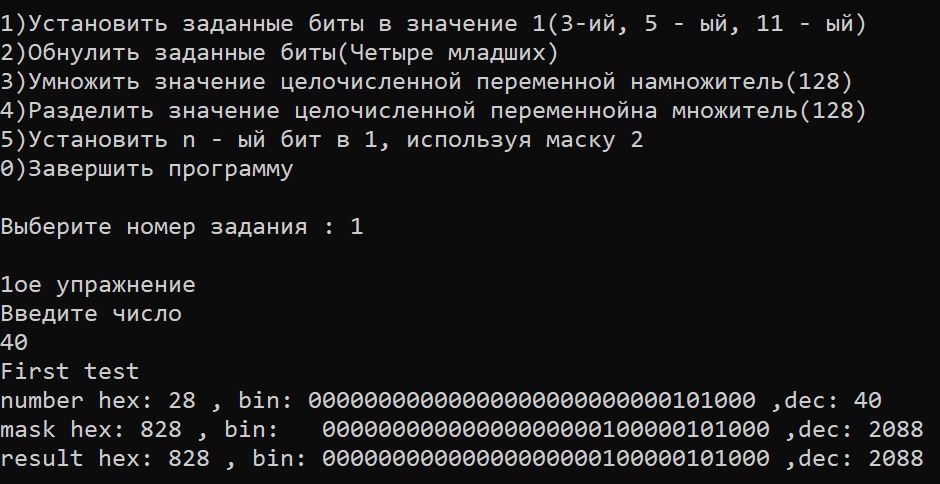


Рисунок 7 - тест 2

# **Вывод**

В результате выполнения работы я:

1. Освоил алгоритмы работы с поразрядными операциями и их реализацию на языке программирования C++
2. Научился программировать автоматическое тестирование простых программ

# **Исходный код программы**

|  |
| --- |
| #include "locale.h"  #include <string>  #include <iostream>    using namespace std;  string convert\_to\_bin(unsigned int num)  {  unsigned int t = 0x80000000;  string result = "";  if (num == 0) return "0";  while (t>0) {    if (t & num) {  result += "1";  }  else {  result += "0";  }  t /= 2;  }  return result;    }    void third(unsigned int& numb, int& digit) {    numb = numb << digit;  }  void fourth(unsigned int& numb, int& digit) {    numb = numb >> digit;    }  void fifth(unsigned int& numb, unsigned int& mask, int& n) {  mask = mask >> n;  numb = numb | mask;  }  int main()  {  setlocale(LC\_ALL, "Rus");  int g = -1;  while (g!=0)  {  cout<<"1)Установить заданные биты в значение 1(3-ий, 5 - ый, 11 - ый)\n2)Обнулить заданные биты(Четыре младших)\n3)Умножить значение целочисленной переменной на множитель(128)\n4)Разделить значение целочисленной переменнойна множитель(128)\n5)Установить n - ый бит в 1, используя маску 2\n0)Завершить программу\n\nВыберите номер задания : ";  cin >> g;  cout << endl;  switch (g) {  case 1: {  cout << "1ое упражнение\n";  unsigned int numb;  cout << "Введите число" << "\n";  cin >> numb;  unsigned int mask = 2088;  cout << "First test\n";  cout << "number hex: " << hex << numb << " , bin: " << convert\_to\_bin(numb) << " ,dec: " << dec << numb << endl;  cout << "mask hex: " << hex << mask << " , bin: " << convert\_to\_bin(mask) << " ,dec: " << dec << mask << endl;  cout << "result hex: " << hex << (numb | mask) << " , bin: " << convert\_to\_bin(numb|mask) << " ,dec: " << dec << (numb | mask) << endl << endl;  break;  }  case 2: {  cout << "2ое упражнение\n";  unsigned int numb;  cout << "Введите число" << "\n";  cin >> numb;  unsigned int mask =0xFFFFFFF0 ;  cout << "number hex: " << hex << numb << " , bin: " << convert\_to\_bin(numb) << " ,dec: " << dec << numb << endl;  cout << "mask hex: " << hex << mask << " , bin: " << convert\_to\_bin(mask) << " ,dec: " << dec << mask << endl;  cout << "result hex: " << hex << (numb & mask) << " , bin: " << convert\_to\_bin(numb & mask) << " ,dec: " << dec << (numb & mask) << endl << endl;  }  case 3: {  cout << "3е упражнение\n";  cout << "Введите исходное число: ";  int digit;  unsigned int numb;  cin >> numb;  digit = 7;  cout << "number before" << endl << "hex: " << hex << numb << " , bin: " << convert\_to\_bin(numb) << " ,dec: " << dec << numb << endl;  third(numb, digit);  cout << "number after" << endl << "hex: " << hex << numb << " , bin: " << convert\_to\_bin(numb) << " ,dec: " << dec << numb << endl << endl;  break;  }  case 4: {  cout << "4ое упражнение\n";  cout << "Введите исходное число: ";  unsigned int numb;  int digit;  cin >> numb;  digit = 7;  cout << "number before" << endl << "hex: " << hex << numb << " , bin: " << convert\_to\_bin(numb) << " ,dec: " << dec << numb << endl;  fourth(numb, digit);  cout << "number after" << endl << "hex: " << hex << numb << " , bin: " << convert\_to\_bin(numb) << " ,dec: " << dec << numb << endl << endl;  break;  }  case 5: {  cout << "5ое упражнение\n";  cout << "Введите исходное число: ";  unsigned int numb;  cin >> numb;  cout << endl;  int n;  cout << "Введите n: ";  cin >> n;  unsigned int mask;  mask = 0x80000000;  cout << "number before" << endl << "hex: " << hex << numb << " , bin: " << convert\_to\_bin(numb) << " ,dec: " << dec << numb << endl;  fifth(numb, mask, n);  cout << "number after" << endl << "hex: " << hex << numb << " , bin: " << convert\_to\_bin(numb) << " ,dec: " << dec << numb << endl << endl;  break;  }  }  }  } |