|  |
| --- |
| https://www.mirea.ru/bitrix/templates/unlimtech/images/logo.png |
| 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 №1  По дисциплине «Структуры и алгоритмы обработки данных» | |
| **Тема:**  Поразрядные операции и их применение | |
|  | |
| Выполнил студент | Сидоров С.Д. |
| группа | ИКБО-20-21 |

Тема. Поразрядные операции и их применение.

Цель. Получить навыки применения поразрядных операций в алгоритмах.

**Задание 1:**

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

**Вариант 3:** Только с четными номерами.

**Исполнение**: Была разработана специальная функция для получения маски необходимой длины, где каждый четный бит установлен в 1 (Листинг 1.1). На входное число применяется маска с помощью | (побитовое или) таким образом заданные биты устанавливаются в 1 (Листинг 1.2).

**int** create\_mask\_ex1(**int** length) {  
 int result = ***0b1***;  
 for (**int** i = 0; i < length/2; i++) {

result = result << 2;

result |= (1 << 0);

result &= ~(1 << 1);

result |= (1 << 0);

}

return result;  
}

Листинг 1.1

**void** exercise\_1(**int** base\_number) {  
  
 int mask = create\_mask\_ex1(decToBin(base\_number).length());

cout << "Exercise 1" << endl;

cout<<"Number: "<<base\_number << " ; " << decToBin(base\_number) << endl;

cout << "Mask: " << mask << " ; " << decToBin(mask) << endl;

cout << "Output: " << (base\_number | mask) << " ; " <<

decToBin(base\_number | mask) << endl;  
}

Листинг 1.2

**Задание 2:**

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

**Вариант 3:** 7-ой, 9-ый, 11-ый

**Исполнение**: Была разработана функция для создания маски необходимой длины где 7-ой, 9-ый, 11-ый биты установлены в 0 (Листинг 2.1). С помощью данной маски и побитового оператора &(и) в заданном на вход числе 7-ой, 9-ый, 11-ый биты устанавливаются в 0 (Листинг 2.2)

**int** create\_mask\_ex2(**int** length) {  
int result = ***0b1***;  
 for (**int** i = 0; i < length; i++) {

result = result << 1;

result |= (1 << 0);

}

result &= ~(1 << 7);

result &= ~(1 << 9);

result &= ~(1 << 11);

return result;  
}

Листинг 2.1

**void** exercise\_2(**int** base\_number) {  
 int mask = create\_mask\_ex2(decToBin(base\_number).length());

cout << "Exercise 2" << endl;

cout << "Number: " << base\_number << " ; " << decToBin(base\_number) << endl;

cout << "Mask: " << mask << " ; " << decToBin(mask) << endl;

cout << "Output: " << (base\_number & mask) << " ; " << decToBin(base\_number & mask) << endl;

}

Листинг 2.2

**Задание 3:**

**Задание**: Определить переменную целого типа. Разработать оператор присваивания и выражение, которое умножает значение переменной на число, указанное в третьем столбце варианта, используя соответствующую поразрядную операцию. Изменяемое число вводится с клавиатуры.

**Вариант 3:** 16

**Исполнение**: С помощью сдвига влево двоичного кода (умножение на 2^n) x << 4 было выполнено умножение числа на 16 == 2^4 (Листинг 3.1)

void exercise\_3(int base\_number) {

cout << "Exercise 3" << endl;

cout << "Number: " << base\_number << " ; " << decToBin(base\_number) << endl;

cout << "Output: " << (base\_number << 4) << " ; " << decToBin(base\_number <<4) << endl;

}

Листинг 3.1

**Задание 4:**

**Задание**: Определить переменную целого типа. Разработать оператор присваивания и выражение, которое умножает значение переменной на число, указанное в третьем столбце варианта, используя соответствующую поразрядную операцию. Изменяемое число вводится с клавиатуры.

**Вариант 3:** 16

**Исполнение**: С помощью сдвига вправо двоичного кода (деление на 2^n) x >> 4 было выполнено умножение числа на 16 == 2^4 (Листинг 4.1)

void exercise\_4(int base\_number) {

cout << "Exercise 4" << endl;

cout << "Number: " << base\_number << " ; " << decToBin(base\_number) << endl;

cout << "Output: " << (base\_number >> 4) << " ; " << decToBin(base\_number >> 4) << endl;

}

Листинг 4.1

**Задание 5:**

**Задание**: 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.

**Вариант 3:** Обнулить n-ый бит в 1, используя маску пункта 1

**Исполнение**: Была создана изначальная маска со значение 0b1, после единица в маске сдвигалась до n-го значения. Для получения необходимого результата применялся X^(исключающее или) base\_number^mask, в таком случае n-ый бит обнулялся. (Листинг 5.1)

void exercise\_5(int base\_number, int n) {

int mask = 0b1;

mask = 1 << n;

cout << "Exercise 5" << endl;

cout << "Number: " << base\_number << " ; " << decToBin(base\_number) << endl;

cout << "Mask: " << (mask) << " ; " << decToBin((mask)) << endl;

cout << "Output: " << (base\_number^(mask)) << " ; " << decToBin(base\_number^(mask));

}

Листинг 5.1

**Применение**:

Была использована дополнительная функция для перевод числа в формат для показа битов (Листинг 6.1), также весь ввод входных данных производился в функции main (Листинг 6.2).

string decToBin(int num)

{

if (num == 0) return "000000000000";

string bin = "";

char tmp;

while (num > 0) {

if ((num & 1) == 0) {

tmp = '0';

}

else {

tmp = '1';

}

bin = tmp + bin;

num = num >> 1;

}

return bin;

}

Листинг 6.1

void main() {

exercise\_1(0xFF0);

exercise\_2(0xFFF0);

int base\_number\_ex\_3;

cout << "Enter number for ex3: ";

cin >> base\_number\_ex\_3;

exercise\_3(base\_number\_ex\_3);

int base\_number\_ex\_4;

cout << "Enter number for ex4: ";

cin >> base\_number\_ex\_4;

exercise\_4(base\_number\_ex\_4);

int base\_number\_ex\_5, count\_bits;

cout << "Enter number for ex5: ";

cin >> base\_number\_ex\_5;

cout << "Enter n for ex5: ";

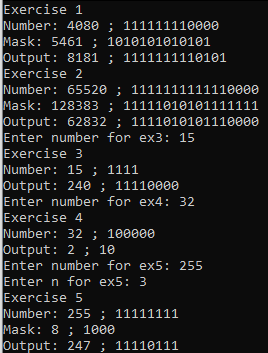
cin >> count\_bits;

exercise\_5(base\_number\_ex\_5, count\_bits);

}

Листинг 6.2

**Тестирование**:



**Выводы**:

В ходе работы были выполнены поставленные задачи и получены навыки программирования на языке C++.В результате выполнения практической работы были получены знания по реализации побитовых операций на примере пяти упражнений.

**Список литературы:**

⦁ Лекции по структурам и алгоритмам обработки данных Рысин М.Л.

⦁ Методическое пособие по выполнению задания 1(битовые операции)