**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра ПИ**

Лабораторная работа № 11

По дисциплине «Основы алгоритмизации и программирования»

На тему «Битовые операции»

Выполнил:  
Студент 1 курса, 10 группа  
Короткевич Артём Сергеевич  
Преподаватель: асс. Харланович А.В

2024, Минск

**Задание 1**  
Изучить использование битовых операций и маскирования числа, опробовав программу, записанную в правой части, с различными исходными числами.  
  
**Код**

#include <iostream>  
using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

unsigned int value; int i;

const unsigned int mask = 1 << 31;

cout << "Введите целое число ";

cin >> value;

cout << "Двоичный вид: ";

for (i = 1; i <= 32; i++)

{

putchar(mask & value ? '1' : '0');

value <<= 1;

if (i % 8 == 0) putchar(' ');

}  
}

**Результат**

  
  
  
  
  
  
  
**Задание 2**  
Выполнить программу, записанную в правой части. Ознакомиться с результатом.  
Опробовать программу, изменяя различные биты различных чисел.

**Код(для числа 100, изменили 2 бита на 3)**

#include <iostream>  
using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int A = 100; char tmp[33]; // массив, который хранит двоичное представление числа

\_itoa\_s(A, tmp, 3); // преобразование целого числа в строку

cout << " Число А: " << tmp << endl;

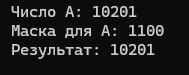
\_itoa\_s(0x24, tmp, 3); // представление числа в 16-ричной системе счисления

cout << " Маска для А: " << tmp << endl;

\_itoa\_s(A | 0x24, tmp, 3);

cout << " Результат: " << tmp << endl << endl;

}

**Результат  
  
  
  
  
Задание 3**В программе, записанной в правой части, используются различные битовые операции.  
Внести изменения в программу с тем, чтобы проверялось число на кратность четырем.  
  
Код

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int A; char tmp[33];

cout << "Введите число ";

cin >> A;

\_itoa\_s(A, tmp, 2);

cout << "Число в двоичном виде = " << tmp << endl;

if ((A & 3) == 0) // внесенное изменение(проверка, равны ли два младших бита числа 0)

cout << "Число кратно 4" << endl;

else

cout << "Число не кратно 4" << endl;

}

Результат  
  
  
  
  
  
  
  
  
**Задание 4**  
В правой части приведен пример программы, демонстрирующей использование битовых операций.  
Проанализировать текст программы и написать пояснения.  
  
**Код + комментарии**

#include <iostream>  
using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

char tmp[33]; // объявление массива символов размером 33 байта

int A, B, maskA = 14;

int maskB = ~maskA >> 1; // создание maskB; инверсия битов maskA; сдвиг 1 бита вправо

cout << "Первое число А="; cin >> A;

cout << "Второе число В="; cin >> B;

\_itoa\_s(A, tmp, 2); // преобразование целого числа в строку в 2-иочой с/c

cout << "A=" << tmp << endl;

\_itoa\_s(B, tmp, 2);

cout << "B=" << tmp << endl;

\_itoa\_s(maskA, tmp, 2);

cout << "Маска для А: " << tmp << endl;

\_itoa\_s((A & maskA) >> 1, tmp, 2); // битовая операция между и А и maskA; сдвиг бита вправо на 1

cout << "Выделенные биты А: " << tmp << endl;

\_itoa\_s(maskB, tmp, 2);

cout << "Маска для В: " << tmp << endl;

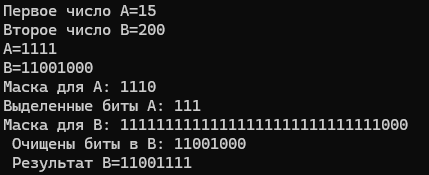
\_itoa\_s(B & maskB, tmp, 2); // // битовая операция между и B и maskB; очистка битов в B на основе maskB

cout << " Очищены биты в B: " << tmp << endl;

\_itoa\_s(((B & maskB) | ((A & maskA) >> 1)), tmp, 2); //объединение битов

cout << " Результат B=" << tmp << endl;

}

**Результат**  
****  
  
  
  
  
  
  
**Задание 5**  
В соответствии со своим вариантом разработать программы, использующие ***битовые*** операции для решения задач, представленных в таблице.  
Результаты одной из программ представить в **Отладчике**.  
  
**Вариант 8**

1. Установить в ноль каждый третий значащий бит целого числа А.
2. Извлечь 3 бита числа **А**, начиная с позиции **n**, и вставить в число **В**, начиная с позиции **m**.

**Код(1)**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_CTYPE, "Russian");

char tmp[33];

int A, mask = 0x92492492; // маска для сброса каждого третьего бита

cout << "Введите число A: ";

cin >> A;

\_itoa\_s(A, tmp, 2);

cout << "A (исходное) = " << tmp << endl;

A = A & ~mask;

\_itoa\_s(A, tmp, 2);

cout << "A (обнуление каждого третьего бита) = " << tmp << endl;  
}

**Результат(1)**

**Примечание: Ошибка в этом коде при выводе в консоли, результат убрал  
  
Код(2)**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

char tmp[33];

int A, B, n, m;

cout << "Введите число A: ";

cin >> A;

cout << "Введите число B: ";

cin >> B;

cout << "Введите позицию n (извлечение из A): ";

cin >> n;

cout << "Введите позицию m (вставка в B): ";

cin >> m;

int maskA = 0b111 << n; // Создаем маску для выделения 3 бит из числа A, начиная с позиции n

int threeBits = (A & maskA) >> n; // Извлекаем 3 бита из числа A

int maskB = ~(0b111 << m); // Создаем маску для очистки 3 бит в числе B, начиная с позиции m

int rezB = (B & maskB) | (threeBits << m);

\_itoa\_s(A, tmp, 2);

cout << "A в двоичном виде: " << tmp << endl;

\_itoa\_s(B, tmp, 2);

cout << "B в двоичном виде: " << tmp << endl;

\_itoa\_s(maskA, tmp, 2);

cout << "Маска для A: " << tmp << endl;

\_itoa\_s(threeBits, tmp, 2);

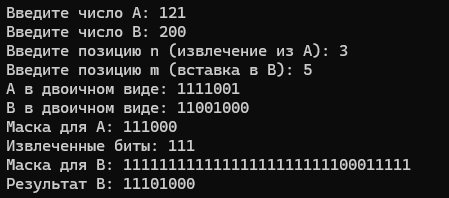
cout << "Извлеченные биты: " << tmp << endl;

\_itoa\_s(maskB, tmp, 2);

cout << "Маска для B: " << tmp << endl;

\_itoa\_s(rezB, tmp, 2);

cout << "Результат B: " << tmp << endl;

} **Результат(2)  
  
**

**Дополнительные задания:  
  
Вариант 15**1. Используя битовые операции проверить, кратно ли двум число **А**.

2. Установить в 0 **n** битов в числе **А** влево от позиции **p**.  
  
**Код(1)**

#include <iostream>  
using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int A;

char tmp[33];

cout << "Введите число: ";

cin >> A;

\_itoa\_s(A, tmp, 2);

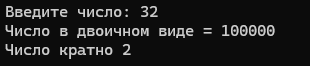
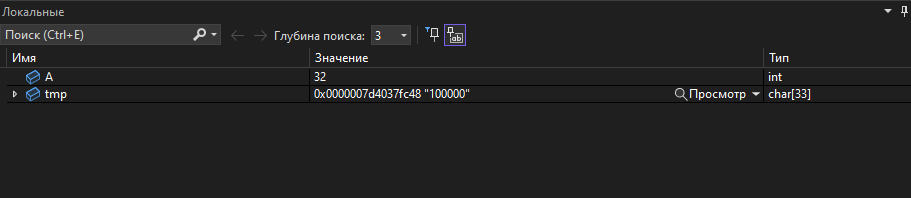
cout << "Число в двоичном виде = " << tmp << endl;

if ((A & 1) == 0)

cout << "Число кратно 2" << endl;

else

cout << "Число не кратно 2" << endl;  
}

**Результат(1)**  
  
  
  
**Отладчик(1)**  
  
  
  
**Код(2)**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int A, p, n;

cout << "Число A: ";

cin >> A;

cout << "Позиция слево: ";

cin >> p;

cout << "Количество битов: ";

cin >> n;

char tmp[33];

\_itoa\_s(A, tmp, 2);

cout << tmp << endl;

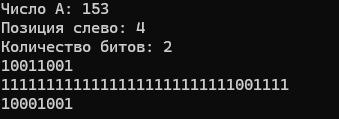
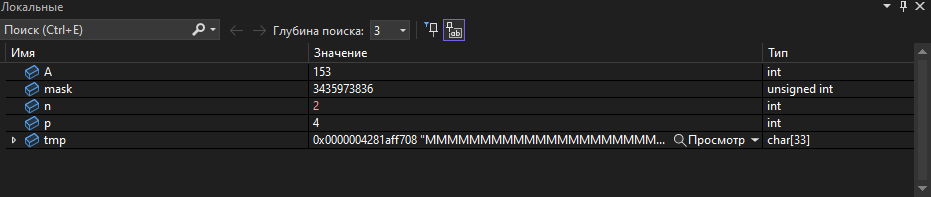
int mask = ~((1 << (p + n)) - (1 << p));

\_itoa\_s(mask, tmp, 2);

cout << tmp << endl;

\_itoa\_s(A & mask, tmp, 2);

cout << tmp << endl;  
}

1. **Результат(2)**  
     
     
     
   **Отладчик(2)  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   Вариант 3**1. Ввести целое число **A**. Инвертировать все биты с 2 по 14, включая эти биты. Вывести результат.

2. Инвертировать **n** битов в числе **А** влево от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**.  
  
**Код(1)**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int A;

char tmp[33];

cout << "Введите число A: ";

cin >> A;

int mask1 = ((1 << 15) - 1) ^ ((1 << 2) - 1); // биты = 1(с 2 по 14)

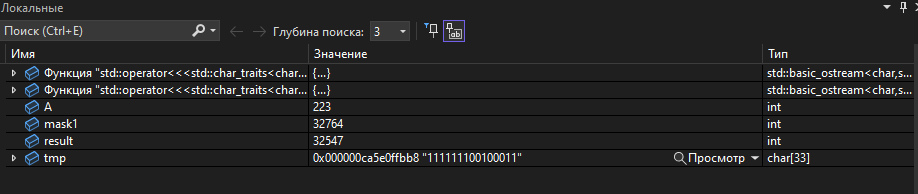
int result = A ^ mask1; // инверсия

\_itoa\_s(A, tmp, 2);

cout << "Число A в двоичном виде: " << tmp << endl;

\_itoa\_s(result, tmp, 2);

cout << "Результат после инверсии битов с 2 по 14: " << tmp << endl;  
}

**Результат(1)**  
  
  
  
**Отладчик(1)**  
  
  
  
**Код(2)**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_CTYPE, "Russian");

int B, p, q, n, m;

char tmp[33];

cout << "Введите число B: "; cin >> B;

cout << "Позиция p (слева): "; cin >> p;

cout << "Количество битов n: "; cin >> n;

cout << "Количество битов m: "; cin >> m;

cout << "Позиция q (в числе B): "; cin >> q;

int mask1 = ((1 << n) - 1) << p; // выбор n битов из числа A начиная с позиции p

int bits = (~B & mask1) >> p; // извлечение инверсированных битов

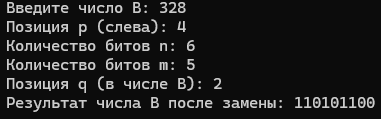
int mask2 = ~(((1 << m) - 1) << q); // очистка m битов в числе B начиная с позиции q

B = B & mask2; // очистка m битов в числе B

B = B | (bits << q); // Вставляем инверсированные биты из числа A в число B

\_itoa\_s(B, tmp, 2);

cout << "Результат числа B после замены: " << tmp << endl;   
}

**Результат(2)**  
  
  
  
  
  
  
**Вариант 5**1. Определить, насколько в числе **А** больше значащих битов, равных единице, чем битов, равных нулю.  
2. Установить в 1 **n** битов в числе **А** вправо от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**.  
  
  
**Код(1)**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int A;

cout << "Число A: ";

cin >> A;

char tmp[33];

\_itoa\_s(A, tmp, 2);

cout << "Число A в двоичном виде: " << tmp << endl;

int bits = 0, ones = 0, zeros = 0;

int foundingBit = 0;

for (int i = 31; i >= 0; --i) { // переход от старшего бита к младшему(всего 32 бита (от 0 до 31))

if (A & (1 << i)) { // проверка, установлен ли текущий бит

foundingBit = 1;

ones++;

}

else if (foundingBit) { // учитываем нули только в значащих битах

zeros++;

}

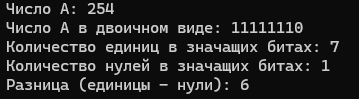
}

int diff = ones - zeros; // разница между 1 и 0

cout << "Количество единиц в значащих битах: " << ones << endl;

cout << "Количество нулей в значащих битах: " << zeros << endl;

cout << "Разница (единицы - нули): " << diff << endl;  
}

**Результат(1)**  
  
  
  
**Код(2)**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

int A, B, p, q, n, m;

char tmp[33];

cout << "Число A: "; cin >> A;

cout << "Число B: "; cin >> B;

cout << "Позиция p (начало битов в A): "; cin >> p;

cout << "Количество бит n в A: "; cin >> n;

cout << "Позиция q (начало битов в B): "; cin >> q;

cout << "Количество бит m в B: "; cin >> m;

\_itoa\_s(A, tmp, 2);

cout << "Число A в двоичном виде: " << tmp << endl;

\_itoa\_s(B, tmp, 2);

cout << "Число B в двоичном виде: " << tmp << endl;

int maskA = ((1 << n) - 1) << p; // установка n битов в A начиная с позиции p

A = A | maskA;

\_itoa\_s(A, tmp, 2);

cout << "Число A после установки битов: " << tmp << endl;

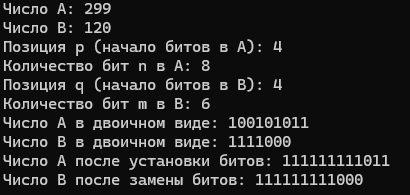
int Bits = (A >> p) & ((1 << n) - 1); // извлечение n битов из A, начиная с позиции p

int maskB = ~(((1 << m) - 1) << q); // очистка m битов в B

B = (B & maskB) | (Bits << q); // установка новых битов

\_itoa\_s(B, tmp, 2);

cout << "Число B после замены битов: " << tmp << endl;

}  
  
**Результат(2)  
  
**  
  
**Отладчик(2)**