Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа №11

По дисциплине «Основы алгоритмизации и программирования»

На тему «Битовые операции»

Выполнил:

Студент 1 курса 7 группы

Ленкевич Павел Андреевич

Преподаватель: асс. Андронова М.В.

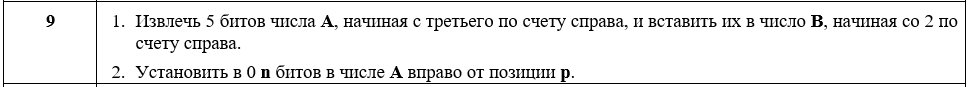
2023, Минск

**Основные задания**

В соответствии со своим вариантом разработать программы, использующие ***битовые*** операции для решения задач, представленных в таблице.

Результаты одной из программ представить в **Отладчике**.

**Вариант 9**

****

**Задание 1.**

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед каждым оператором ввода-вывода

void main()

{

setlocale(LC\_CTYPE, "Russian"); // Подключение русского языка

int A, B; // Подключение переменных

char tmp[33]; // Объявление массива

cout << "Введите A: "; cin >> A; // Вывод текста и ввод значений переменных

cout << "Введите B: "; cin >> B;

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "А в двоичном коде: " << tmp << endl;

\_itoa\_s(B, tmp, 2); // B переводится в двоичную систему и помещается в массив

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

int maskaA = 0b11111 << 2; // Создаётся маски для A

\_itoa\_s(maskaA, tmp, 2); // Маска переводится в двоичный код и помешается в массив

int bitsA = (A & maskaA) >> 2; // Извлекаются биты из А, а затем выполняется сдвиг в право на 2

\_itoa\_s(bitsA, tmp, 2); // bitsA, переводится в двоичную систему и помещается в массив

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

int maskaB = 0b11111 << 1; // Создаётся маски для B

\_itoa\_s(maskaB, tmp, 2); // Маска переводится в двоичный код и помешается в массив

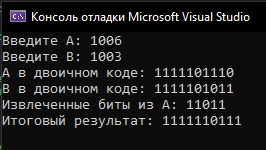
B &= ~maskaB; // Сброс битов В с помощью "и не"

B |= (bitsA << 1); // Вставка битов из А в В с помощью "или"

\_itoa\_s(B, tmp, 2);// Преобразуем В после вставки битов

cout << "Итоговый результат: " << tmp << endl;

}

****

**Задание 2.**

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед каждым оператором ввода-вывода

void main()

{

setlocale(LC\_CTYPE, "Russian"); // Подключение русского языка

int A, n, p; // Инициализация переменных

char tmp[33]; // Создаём массив

cout << " Введите A: "; cin >> A; // Вывод текста и ввод значений переменной

cout << "Введите количество устанавливаемых битов в 0; n: "; cin >> n;

cout << "Введите позицию, с которой устанавливать биты в 0; p: "; cin >> p;

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "А в двоичном коде: " << tmp << endl;

int maskaA = ~(((1 << n) - 1) << p); // Создаётся маска для A начиная с элемента p

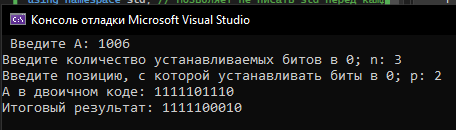
\_itoa\_s(maskaA, tmp, 2); // Маска переводится в двоичный код и помешается в массив

A &= maskaA; // Применяем маску к A с помощью "и"

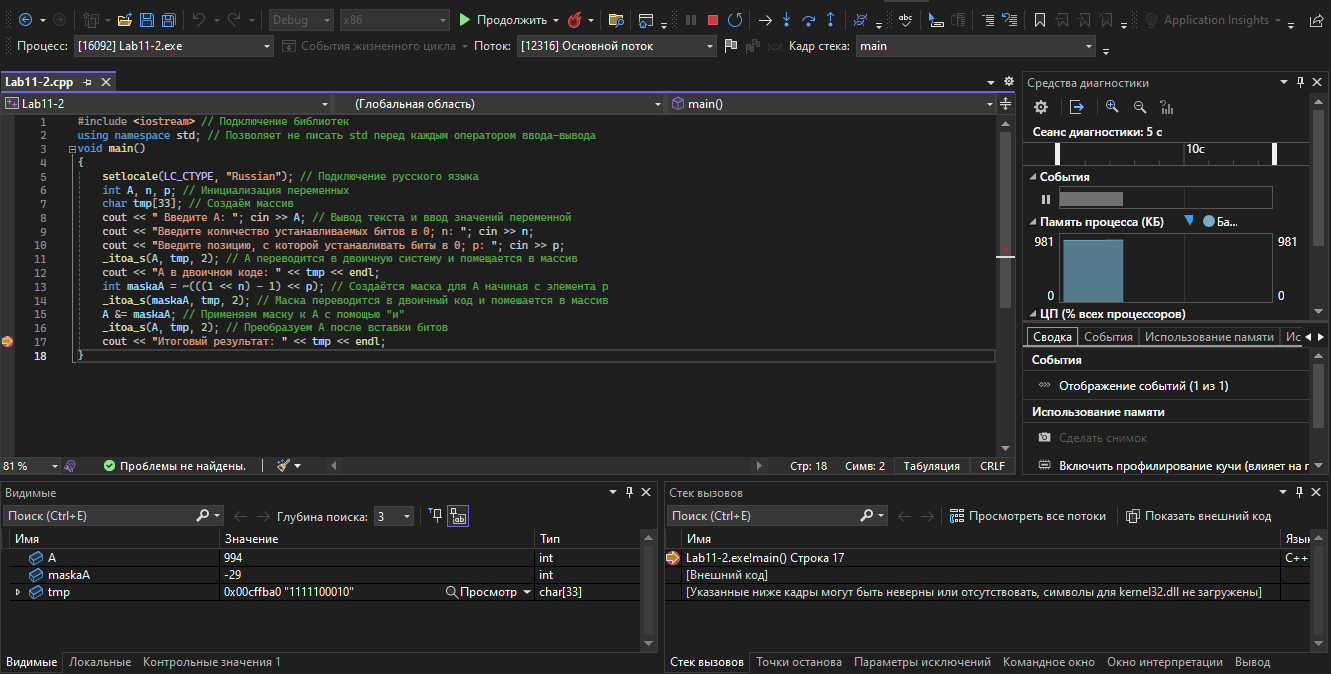
\_itoa\_s(A, tmp, 2); // Преобразуем А после вставки битов

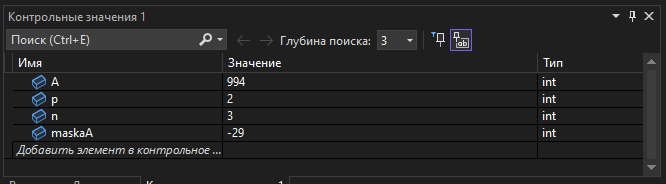
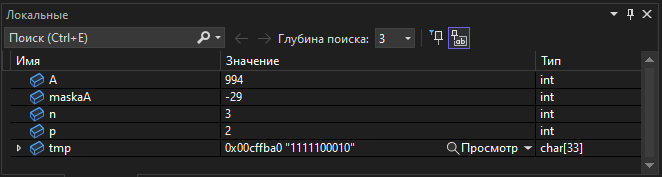
cout << "Итоговый результат: " << tmp << endl;

}

****

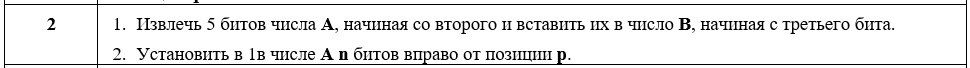
**Отладка:**

****

****

**Дополнительные задания**

**1. Вариант 2**

****

**Задание 1.**

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед каждым оператором ввода-вывода

void main()

{

setlocale(LC\_CTYPE, "Russian"); // Подключение русского языка

int A, B; // Подключение переменных

char tmp[33]; // Объявдение массива

cout << "Введите A: "; cin >> A; // Вывод текста и ввод значений переменных

cout << "Введите B: "; cin >> B;

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "А в двоичном коде: " << tmp << endl;

\_itoa\_s(B, tmp, 2); // B переводится в двоичную систему и помещается в массив

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

int maskaA = 0b11111 << 1; // Создаётся маски для A

\_itoa\_s(maskaA, tmp, 2); // Маска переводится в двоичный код и помешается в массив

int bitsA = (A & maskaA) >> 1; // Извлекаются биты из А, а затем выполняется сдвиг в право на 2

\_itoa\_s(bitsA, tmp, 2); // bitsA, переводится в двоичную систему и помещается в массив

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

int maskaB = 0b11111 << 2; // Создаётся маски для B

\_itoa\_s(maskaB, tmp, 2); // Маска переводится в двоичный код и помешается в массив

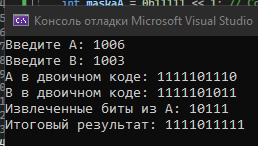
B &= ~maskaB; // Сброс битов В с помощью "и не"

B |= (bitsA << 2); // Вставка битов из А в В с помощьюы "или"

\_itoa\_s(B, tmp, 2);// Преобразуем В после вставки битов

cout << "Итоговый результат: " << tmp << endl;

}

****

**Задание 2.**

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед каждым оператором ввода-вывода

void main()

{

setlocale(LC\_CTYPE, "Russian"); // Подключение русского языка

int A, n, p; // Инициализация переменных

char tmp[33]; // Создаём массив

cout << " Введите A: "; cin >> A; // Вывод текста и ввод значений переменной

cout << "Введите количество устанавливаемых битов в 1; n: "; cin >> n;

cout << "Введите позицию, с которой устанавливать биты в 1; p: "; cin >> p;

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "А в двоичном коде: " << tmp << endl;

int maskaA = (((1 << n) - 1) << p); // Создаётся маска для A начиная с элемента p

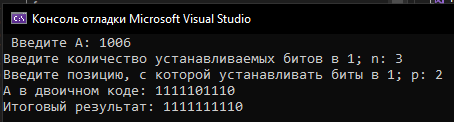
\_itoa\_s(maskaA, tmp, 2); // Маска переводится в двоичный код и помешается в массив

A |= maskaA; // Применяем маску к A с помощью "или"

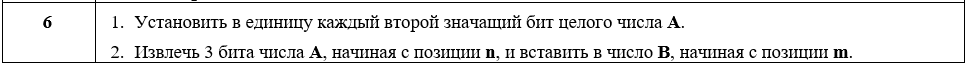
\_itoa\_s(A, tmp, 2); // Преобразуем А после вставки битов

cout << "Итоговый результат: " << tmp << endl;

}

****

**2. Вариант 6**

****

**Задание 1.**

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед каждым оператором ввода-вывода

void main()

{

setlocale(LC\_CTYPE, "Russian"); // Подключение русского языка

int A; // Подключение переменных

char tmp[33]; // Объявление массива

cout << "Введите A: "; cin >> A; // Вывод текста и ввод значений переменных

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "А в двоичном коде: " << tmp << endl;

int maskaA = 0b01010; // Создаётся маски для A

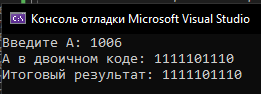
\_itoa\_s(maskaA, tmp, 2); // Маска переводится в двоичный код и помешается в массив

A |= maskaA; // Применение маски на А

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "Итоговый результат: " << tmp << endl;

}

****

**Задание 2.**

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед каждым оператором ввода-вывода

void main()

{

setlocale(LC\_CTYPE, "Russian"); // Подключение русского языка

int A, B, n, m; // Подключение переменных

char tmp[33]; // Объявление массива

cout << "Введите A: "; cin >> A; // Вывод текста и ввод значений переменных

cout << "Введите B: "; cin >> B;

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "А в двоичном коде: " << tmp << endl;

\_itoa\_s(B, tmp, 2); // B переводится в двоичную систему и помещается в массив

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

cout << "Введите начальную позицию для А; n: "; cin >> n;

cout << "Введите начальную позицию для B; m: "; cin >> m;

int maskaA = ((1 << 3) - 1) << n; // Создаётся маски для A

\_itoa\_s(maskaA, tmp, 2); // Маска переводится в двоичный код и помешается в массив

int bitsA = (A & maskaA) >> n; // Извлекаются биты из А, а затем выполняется сдвиг в право на 2

\_itoa\_s(bitsA, tmp, 2); // bitsA, переводится в двоичную систему и помещается в массив

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

int maskaB = ((1 << 3) - 1) << n; // Создаётся маски для B

\_itoa\_s(maskaB, tmp, 2); // Маска переводится в двоичный код и помешается в массив

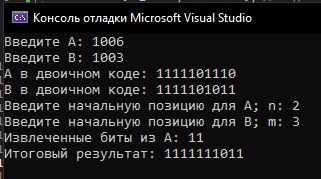
B &= ~maskaB; // Сброс битов В с помощью "и не"

B |= (bitsA << m); // Вставка битов из А в В с помощьюы "или"

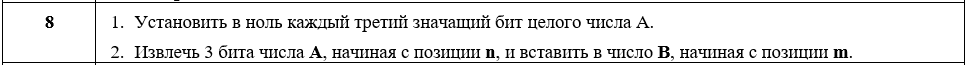
\_itoa\_s(B, tmp, 2);// Преобразуем В после вставки битов

cout << "Итоговый результат: " << tmp << endl << endl;

}

****

**3. Вариант 8**

****

**Задание 1.**

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед каждым оператором ввода-вывода

void main()

{

setlocale(LC\_CTYPE, "Russian"); // Подключение русского языка

int A; // Подключение переменных

char tmp[33]; // Объявдение массива

cout << "Введите A: "; cin >> A; // Вывод текста и ввод значений переменных

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "А в двоичном коде: " << tmp << endl;

int maskaA = 0b011011011011011011; // Создаётся маски для A

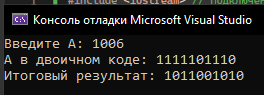
\_itoa\_s(maskaA, tmp, 2); // Маска переводится в двоичный код и помешается в массив

A &= maskaA; // Применение маски на А с помощью операции "и"

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "Итоговый результат: " << tmp << endl;

}

****

**Задание 2.**

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед каждым оператором ввода-вывода

void main()

{

setlocale(LC\_CTYPE, "Russian"); // Подключение русского языка

int A, n, p; // Инициализация переменных

char tmp[33]; // Создаём массив

cout << " Введите A: "; cin >> A; // Вывод текста и ввод значений переменной

cout << "Введите количество устанавливаемых битов в 0; n: "; cin >> n;

cout << "Введите позицию, с которой устанавливать биты в 0; p: "; cin >> p;

\_itoa\_s(A, tmp, 2); // A переводится в двоичную систему и помещается в массив

cout << "А в двоичном коде: " << tmp << endl;

int maskaA = (((1 << n) - 1) << p); // Создаётся маска для A начиная с элемента p

\_itoa\_s(maskaA, tmp, 2); // Маска переводится в двоичный код и помешается в массив

A |= maskaA; // Применяем маску к A с помощью "или"

\_itoa\_s(A, tmp, 2); // Преобразуем А после вставки битов

cout << "Итоговый результат: " << tmp << endl;

}

