**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра программной инженерии**

Лабораторная работа 11

По дисциплине «Основы алгоритмизации и программирования»

На тему «Битовые операции»

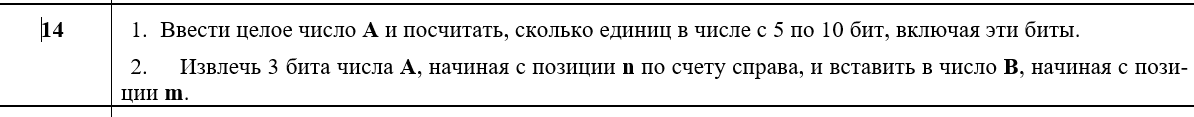
Выполнила:

Студентка 1 курса 7 группы

Шинкевич Марина Дмитриевна

Преподаватель: асс. Андронова М.В.

2023, Минск



#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int A, count = 0;

cout << "Введите целое число: ";

cin >> A;

// сдвигаем число на 5 бит вправо, чтобы биты с 5 по 10 стали первыми шестью битами

A = A >> 5;

// используем маску 0b111111(или 63 в десятичной системе) , чтобы оставить только первые шесть битов

A = A & 0b111111;

// считаем количество единиц в числе

while (A > 0) {

if (A & 1) { // если последний бит равен 1

count++; // увеличиваем счетчик единиц

}

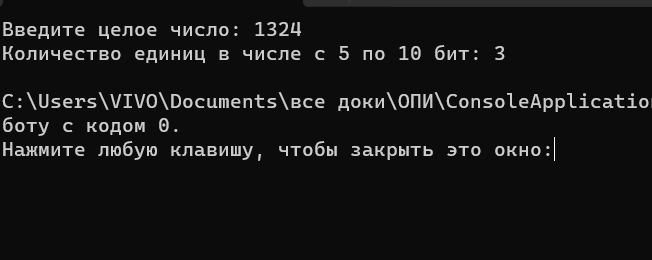
A = A >> 1; // сдвигаем число на один бит вправо

}

cout << "Количество единиц в числе с 5 по 10 бит: " << count << endl;

return 0;

}



#include <iostream>

int main() {

setlocale(LC\_CTYPE, "Russian");

int A, B;

int n, m;

// ввод чисел A и B

std::cout << "Введите число A: ";

std::cin >> A;

std::cout << "Введите число B: ";

std::cin >> B;

// ввод позиции n и m

std::cout << "Введите позицию n: ";

std::cin >> n;

std::cout << "Введите позицию m: ";

std::cin >> m;

int bits = (A >> (n - 1)) & 7;//сдвиг числа А на (n-1) битов вправо с помощью оператора >>. Затем с помощью побитового оператора & извлекаем 3 бита из полученного числа, которые находятся в позициях n, n+1 и n+2. Для этого используется маска 7, которая представляет собой число 111 в двоичной системе.

B &= ~(((1 << 3) - 1) << (m - 1));//сброс трех битов числа B, начиная с позиции m. Для этого создается маска из трех единиц, которая затем с помощью побитового оператора ~ инвертируется, чтобы получить маску из трех нулей. Затем эта маска умножается на 2 в степени (m-1), чтобы получить необходимую позицию для сброса битов.

// вставка выделенных битов в число B, начиная с позиции m

B |= bits << (m - 1);

// вывод числа B с вставленными битами

std::cout << "Число B с вставленными битами: " << B << std::endl;

return 0;

}

