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 6 группы

Кравченко Сергей Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 8

|  |  |
| --- | --- |
| **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**. |

1.

#include <iostream> //паодключение библиотек//

using namespace std; //пространство имен//

int main() {

setlocale(LC\_ALL, "RUS"); //устанавливаем локаль//

int A; //переменная//

cout << "Введите число: ";

cin >> A;

for (int i = 0; i < 32; i += 3) //цикл проходится по каждому третьему биту//Цикл продолжается до 32, потому что в большинстве современных компьютеров размер целого числа составляет 32 бита.

{

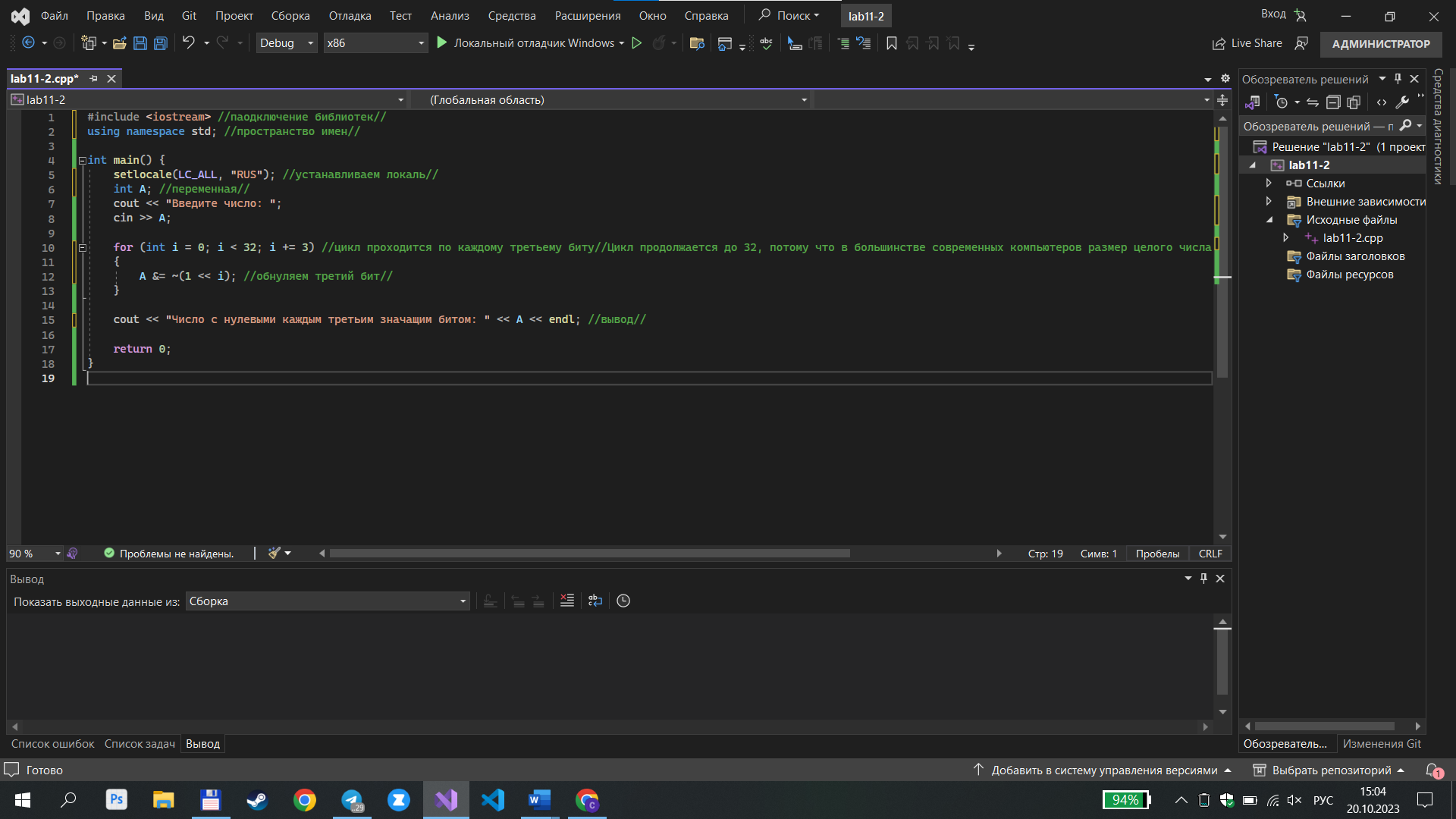
A &= ~(1 << i); //обнуляем третий бит//

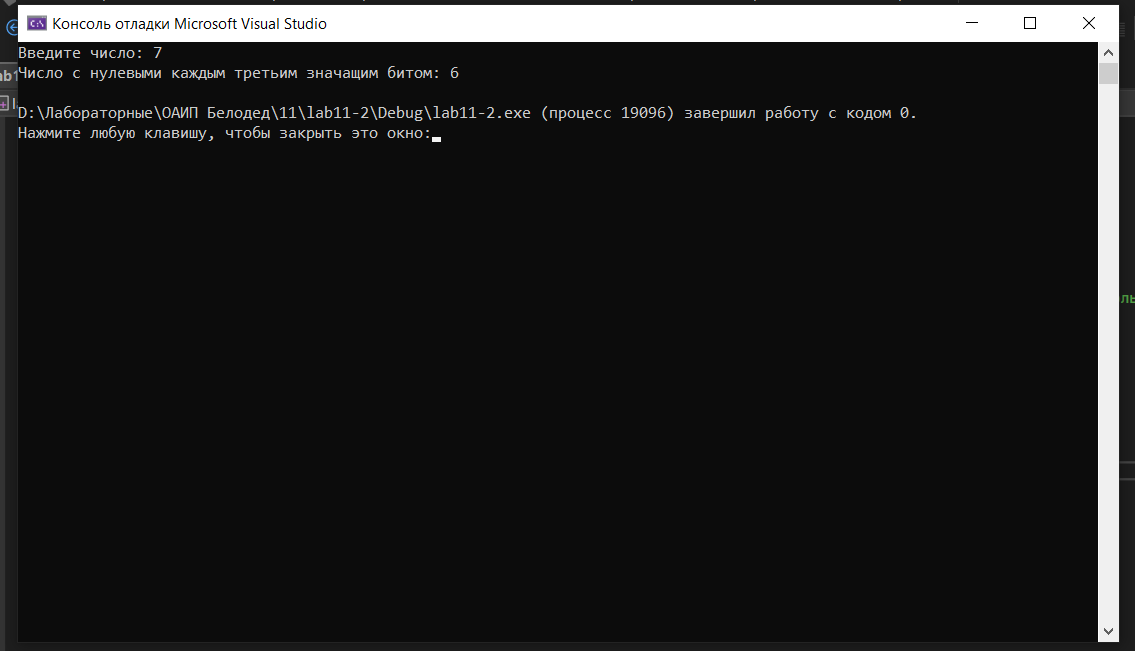
}

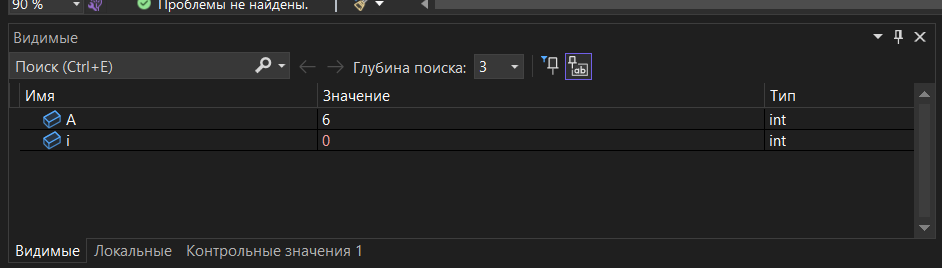
cout << "Число с нулевыми каждым третьим значащим битом: " << A << endl; //вывод//

return 0;

}







2.

#include <iostream> //подключение библиотек//

#include <Windows.h>

using namespace std; //использование пространства имен//

int main() {

SetConsoleOutputCP(1251);

int a, b, l=3,n=0,m=0; //переменные//

char tmp[33]; //массив//

cout << "Введите число A: "; cin >> a;

cout << "Ведите с какой позиции k возьмем биты из числа А: "; cin >> n;

int maskA = 1;

\_itoa\_s(maskA, tmp, 2); //создание маску для числа A//

maskA = maskA << l; //сдвиг маски A для вычисления 3 битов//

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

cout << "maskA << l" << " (l = " << l << "): " << tmp << '\n';

maskA = maskA - 1; //отнимаем 1//

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

maskA = maskA << n; //сдвигаем маску на нужную нам позицию//

cout << "maskA << n" << "(n = " << n << "): " << tmp << '\n'; //вывод сдвинутой маски//

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

cout << "A: " << tmp << '\n';

\_itoa\_s((maskA & a), tmp, 2); //используем битовую операцию и для вычисления 3 битов числа A//

cout << "maskA & A: " << tmp << '\n'; //вывод//

\_itoa\_s((maskA & a) >> n, tmp, 2);

cout << "Введите число B: "; cin >> b;//ввод числа B//

cout << "Введите позицию, с которой будут заменены биты битами из числа А: "; cin >> m; //ввод позиции m//

\_itoa\_s(b, tmp, 2); //переводим число B в двоичную систему//

cout << "B: " << tmp << '\n';

int maskB = 1;

\_itoa\_s(maskB, tmp, 2); //создание маски для числа B//

maskB = maskB << l; //сдвигаем маску B//

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

cout << "maskB << l: " << tmp << '\n';//вывод//

maskB -= 1; //отнимаем 1//

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

maskB = maskB << m; //сдвигаем маску B на позицию m..

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

maskB = ~maskB; //делаем инверсию маски B//

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

\_itoa\_s(maskB & b, tmp, 2); //используем битовую операцию и//

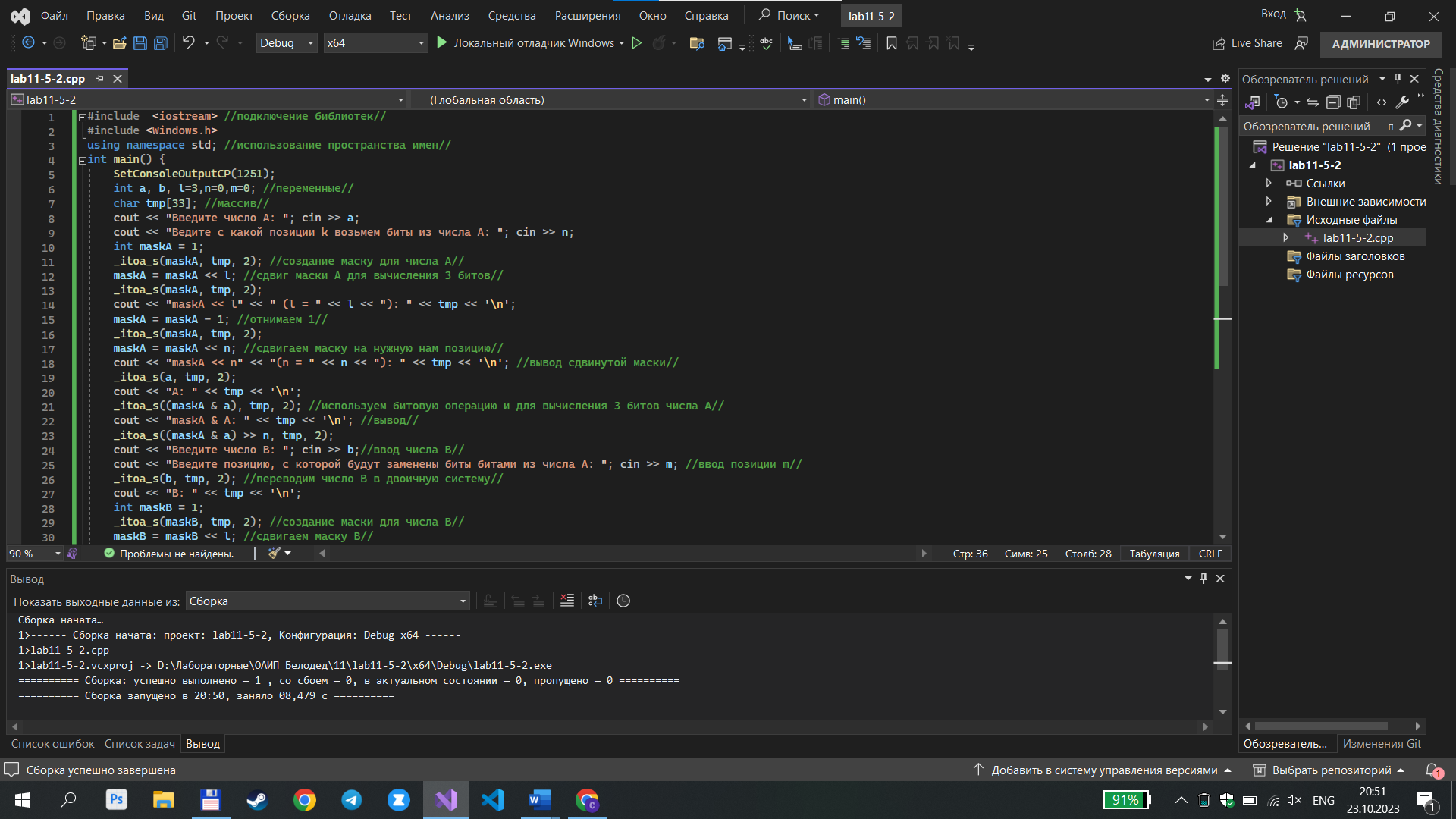
\_itoa\_s(((maskA & a) >> n) << m, tmp, 2); //сдвигаем биты числа A на n вправо и m влево//

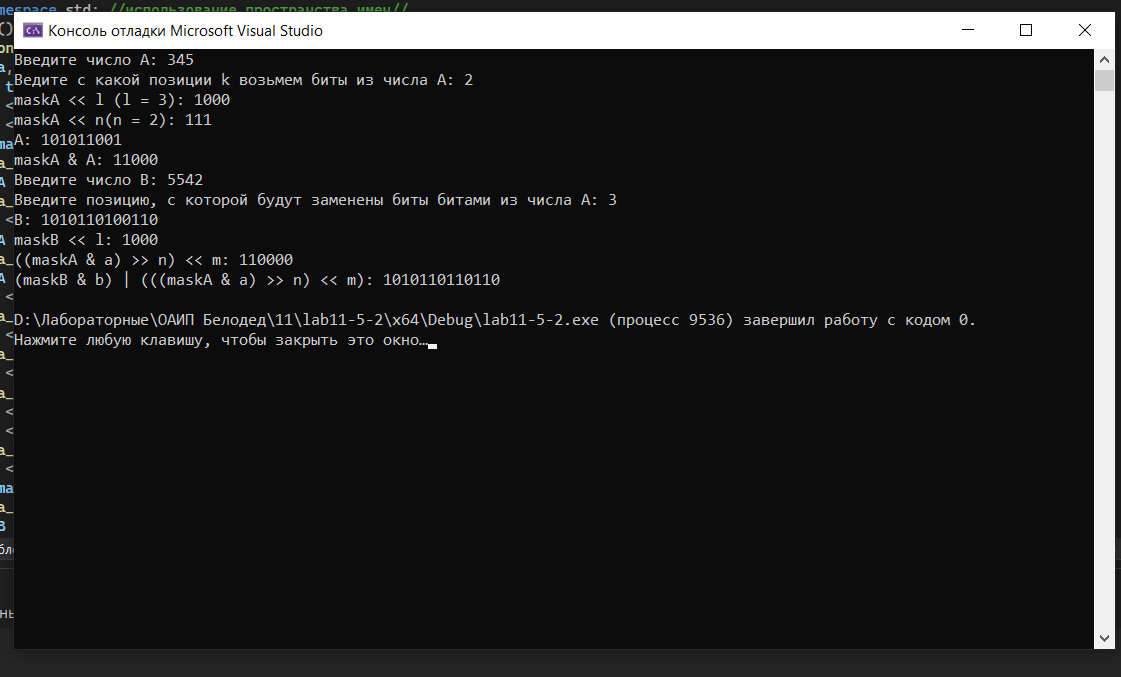
cout << "((maskA & a) >> n) << m: " << tmp << '\n';

\_itoa\_s((maskB & b) | (((maskA & a) >> n) << m), tmp, 2); //перемещаем биты числа A в число B//

cout << "(maskB & b) | (((maskA & a) >> n) << m): " << tmp << '\n'; //выводим конечный результат//

}





Дополнительные варианты

Вариант 10

|  |  |
| --- | --- |
| **10** | 1. Вывести 6 бит целого числа **А**, начиная со 2-ого n битов. 2. Инвертировать **n** битов в числе **А** влево от позиции **p**. |

1.

#include <iostream> //подключение библиотеки//

using namespace std;//использование пространства имен//

int main()

{

setlocale(LC\_ALL, "RUS"); //утсанавливаем локаль//

char tmp[33]; //создаем массив//

int A; //создаем переменные

cout << "Первое число А="; cin >> A;

\_itoa\_s(A, tmp, 2); //перевод данных в двоичную систему счисления

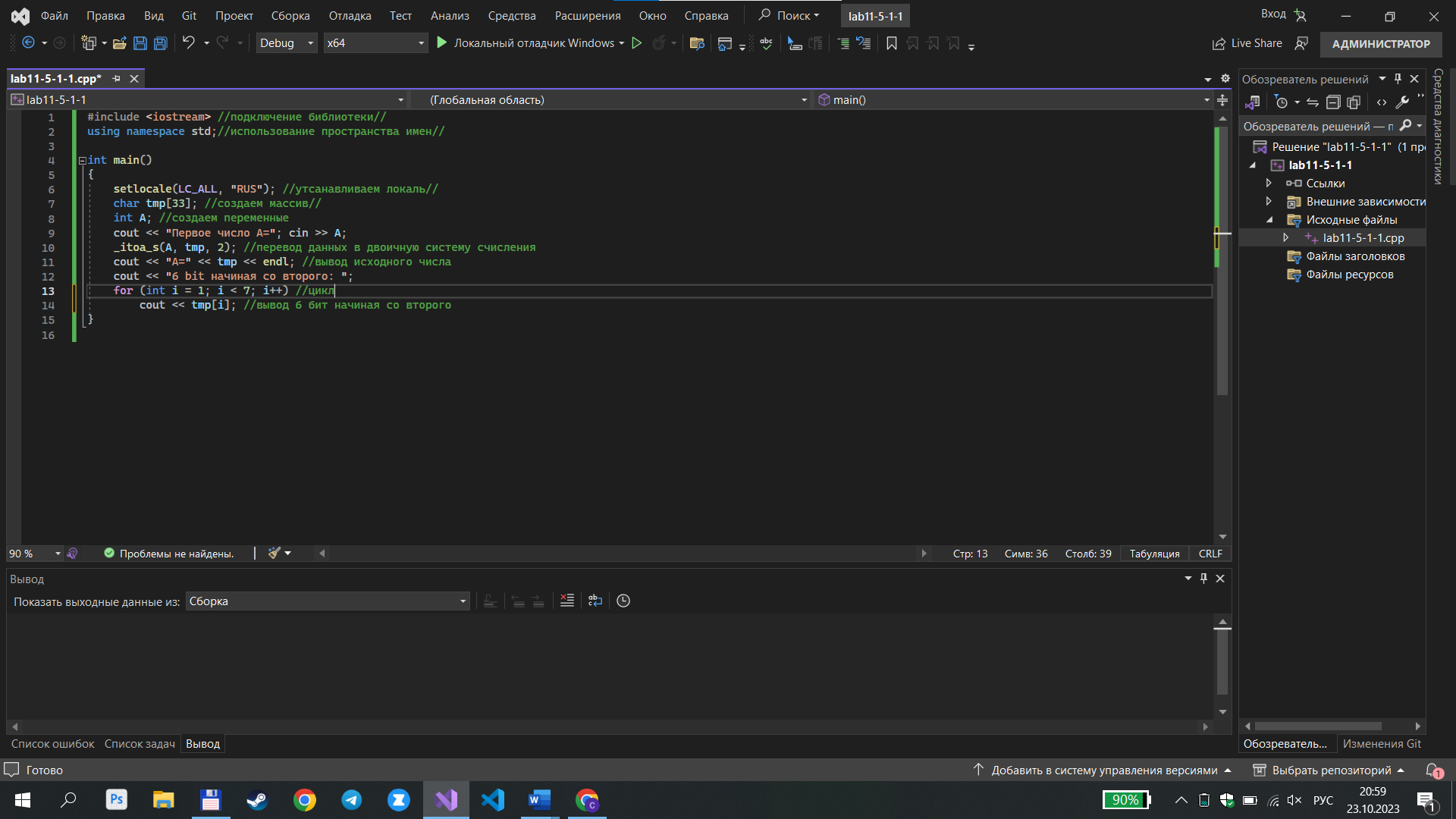
cout << "A=" << tmp << endl; //вывод исходного числа

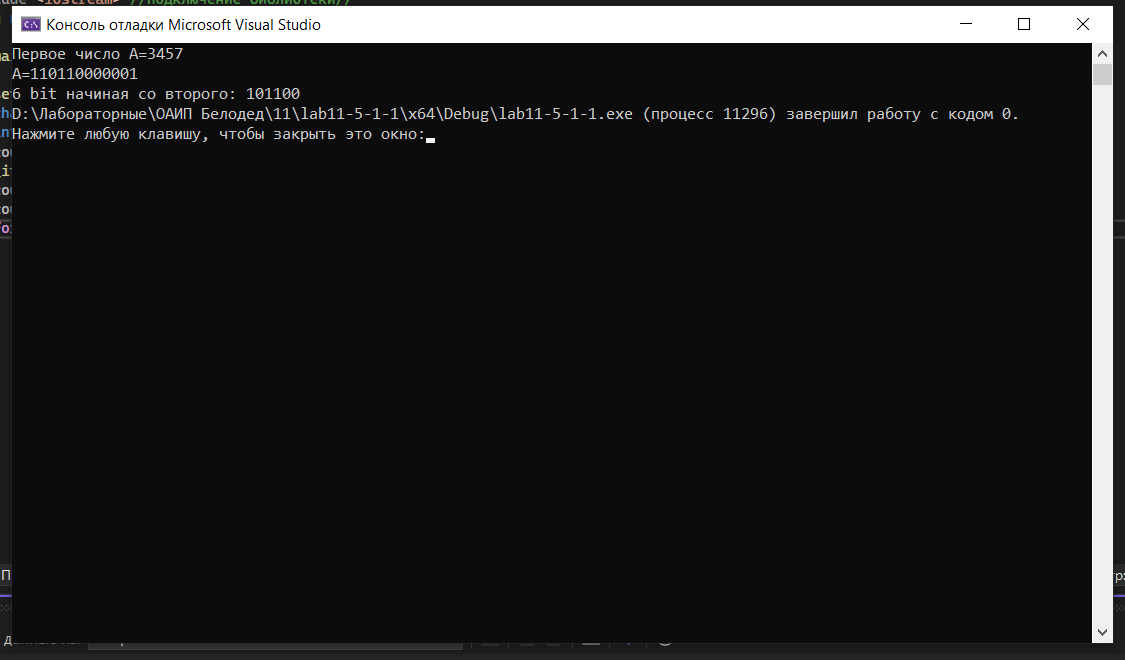
cout << "6 bit начиная со второго: ";

for (int i = 1; i < 7; i++) //цикл

cout << tmp[i]; //вывод 6 бит начиная со второго

}





2.

#include <iostream> //подлкючение библиотек//

using namespace std; //использование пространства имен//

int main()

{

int A, p, n; //переменные//

setlocale(LC\_ALL, "RUS");

cout << "Введите число A: "; //ввод//

cin >> A;

cout << "Введите позицию p: "; //ввод//

cin >> p;

cout << "Введите количество битов n: "; //ввод//

cin >> n;

int mask = (1 << n) - 1; // Создаем маску из n единиц

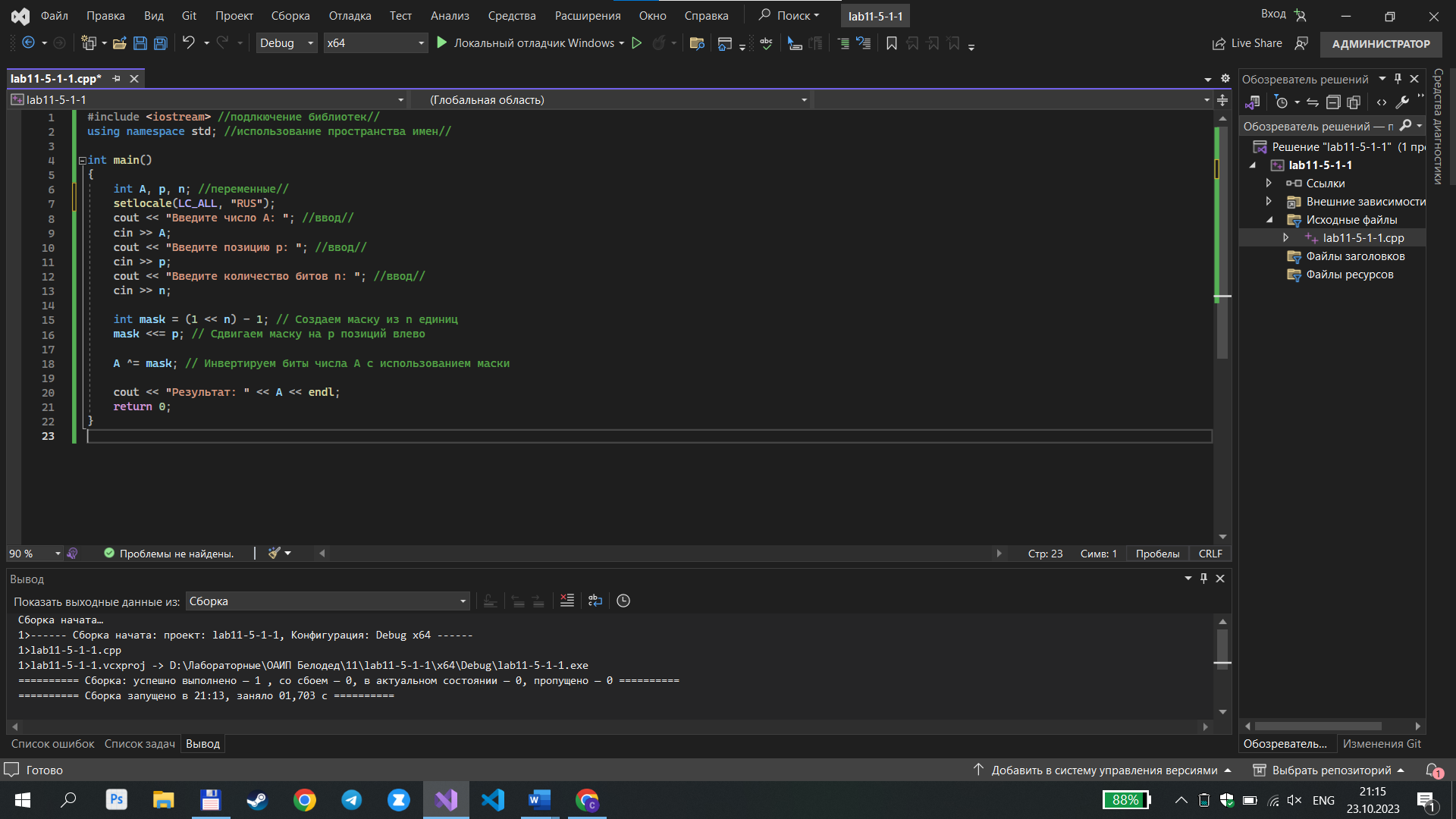
mask <<= p; // Сдвигаем маску на p позиций влево

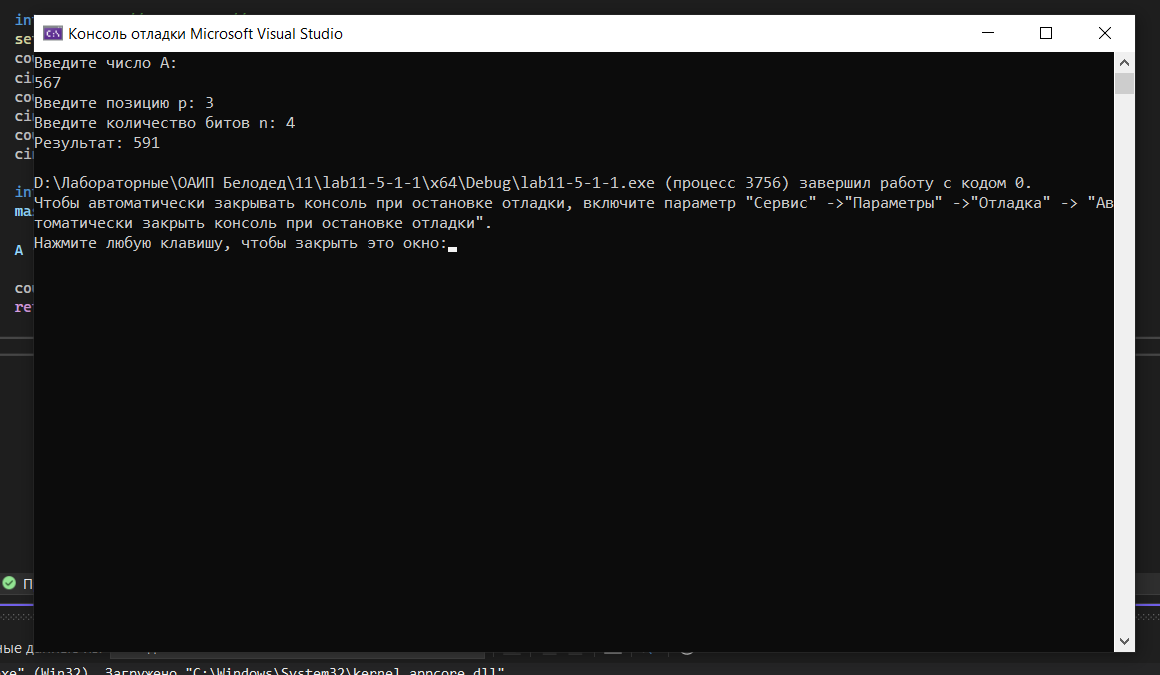
A ^= mask; // Инвертируем биты числа A с использованием маски

cout << "Результат: " << A << endl;

return 0;

}





Вариант 4

|  |  |
| --- | --- |
| **4** | 1. Используя битовые операции проверить, кратно ли четырем число **А**. 2. Установить в 1 в числе **А n** битов влево от позиции **p**. |

1.

#include <iostream>//подключение библиотек//

using namespace std;//использование пространства имен//

int main() {

setlocale(LC\_ALL, "RUS");

int A; //переменная A//

cout << "Введите число: ";

cin >> A;

if ((A & 3) == 0) //проверка условия кратности 4 используя битовую операцию AND//

{

cout << "Число " << A << " кратно четырем." << endl; //вывод//

}

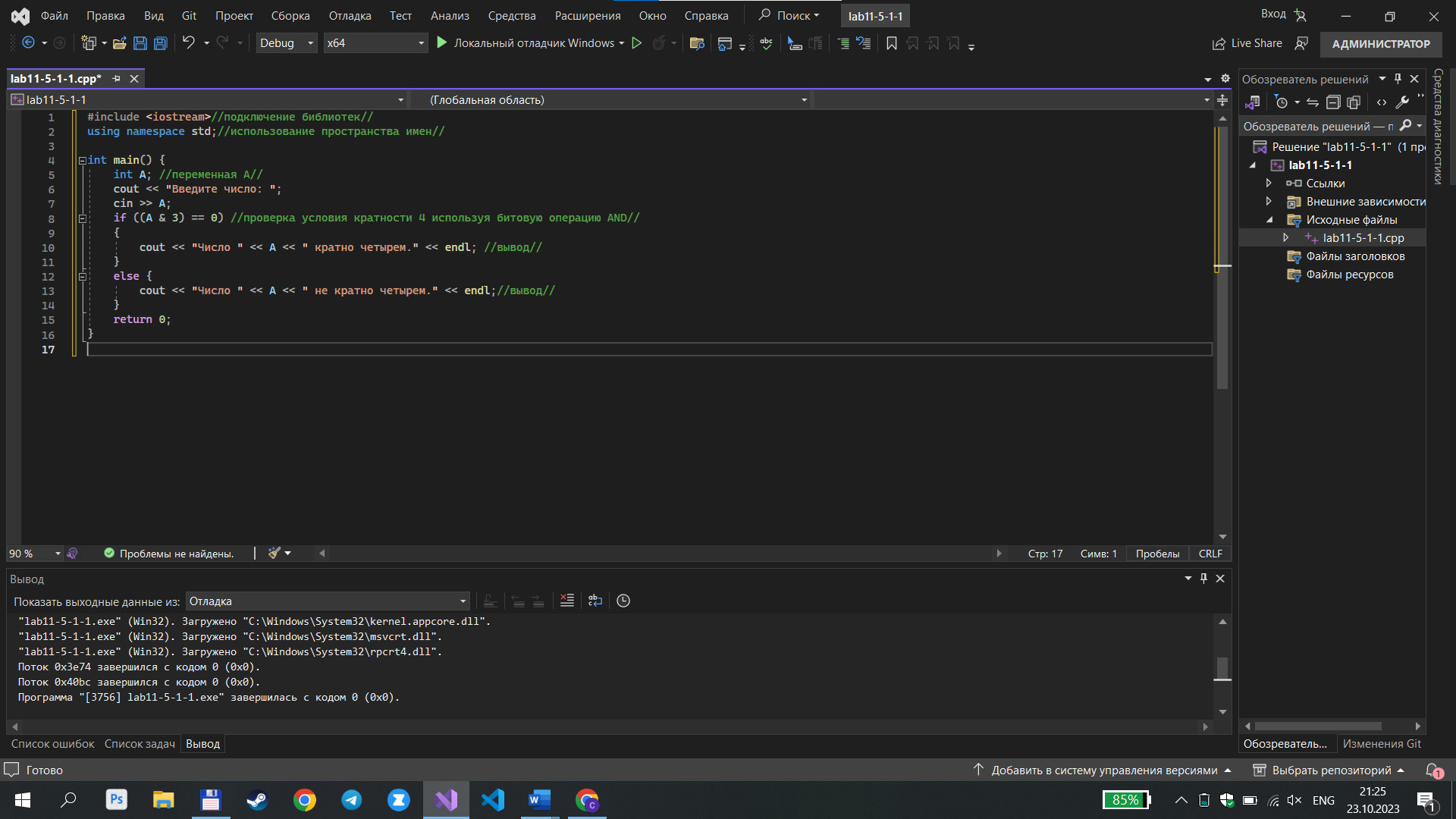
else {

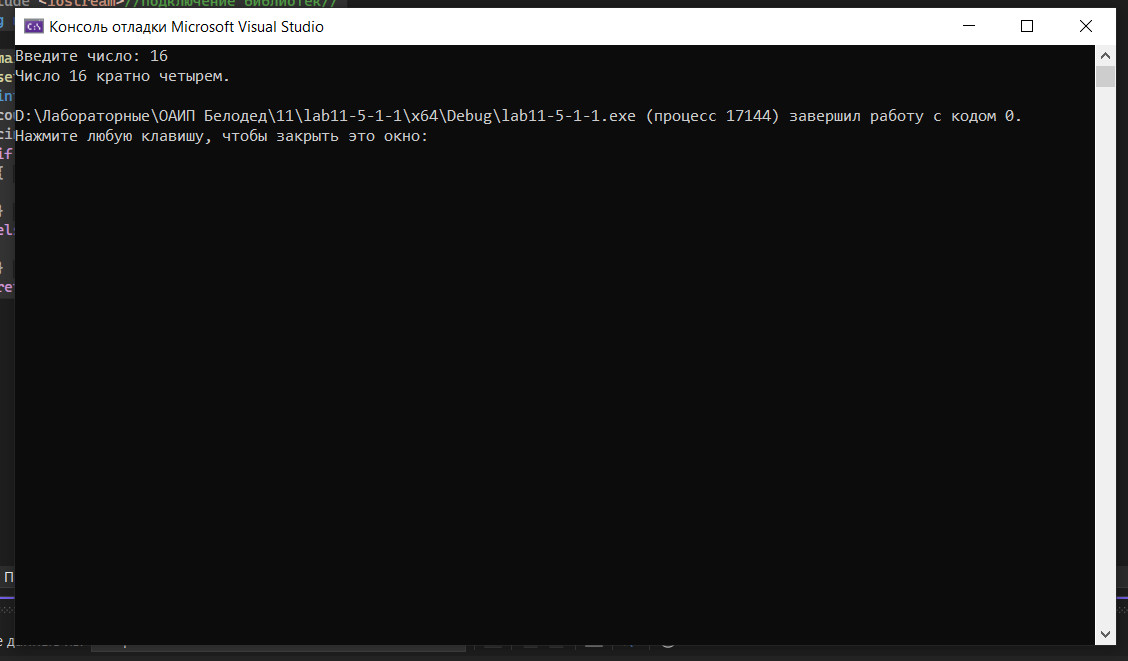
cout << "Число " << A << " не кратно четырем." << endl;//вывод//

}

return 0;

}





Вариант 2

|  |  |
| --- | --- |
| **2** | 1. Извлечь 5 битов числа **A**, начиная со второго и вставить их в число **B**, начиная с третьего бита. 2. Установить в 1в числе **А n** битов вправо от позиции **p**. |

1.

#include <iostream> //подлкючение библиотек//

#include <Windows.h>

using namespace std; //использование пространства имен//

int main() {

SetConsoleOutputCP(1251);

int a, b, n, k, p; //переменные//

char tmp[33]; //массив//

cout << "Введите число A: "; cin >> a;

cout << "Ведите сколько n битов из числа А: "; cin >> n;

cout << "Ведите с какой позиции k возьмем биты из числа А: "; cin >> k;

int maskA = 1; //создание маски

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

maskA = maskA << n; //сдвиг маски A влево на n..

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

maskA = maskA - 1;//отнимаем 1 от маски

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

maskA = maskA << k; //сдвигаем маску A на k влево//

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

cout << "maskA << k" << "(k = " << k << "): " << tmp << '\n';

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

cout << "A: " << tmp << '\n';

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

cout << "maskA & A: " << tmp << '\n'; //маска A после выполнения битовой операции и//

\_itoa\_s((maskA & a) >> k, tmp, 2); //сдвиг маски вправо на k//

cout << "(maskA & A) >> k: " << tmp << '\n';

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

cout << "Введите позицию, с которой будут заменены биты битами из числа А: "; cin >> p;

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

cout << "B: " << tmp << '\n';

int maskB = 1; //создание маски B//

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

maskB = maskB << n; //сдвиг маски B на n влево//

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

maskB -= 1; //отнимаем 1 от маски

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

maskB = maskB << p; //сдвиг маски B на p влево//

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

cout << "maskB << p (p = " << p << " ): " << endl;

maskB = ~maskB; //инвертируем маску//

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

\_itoa\_s(maskB & b, tmp, 2); //выполнение битоаой операции и//

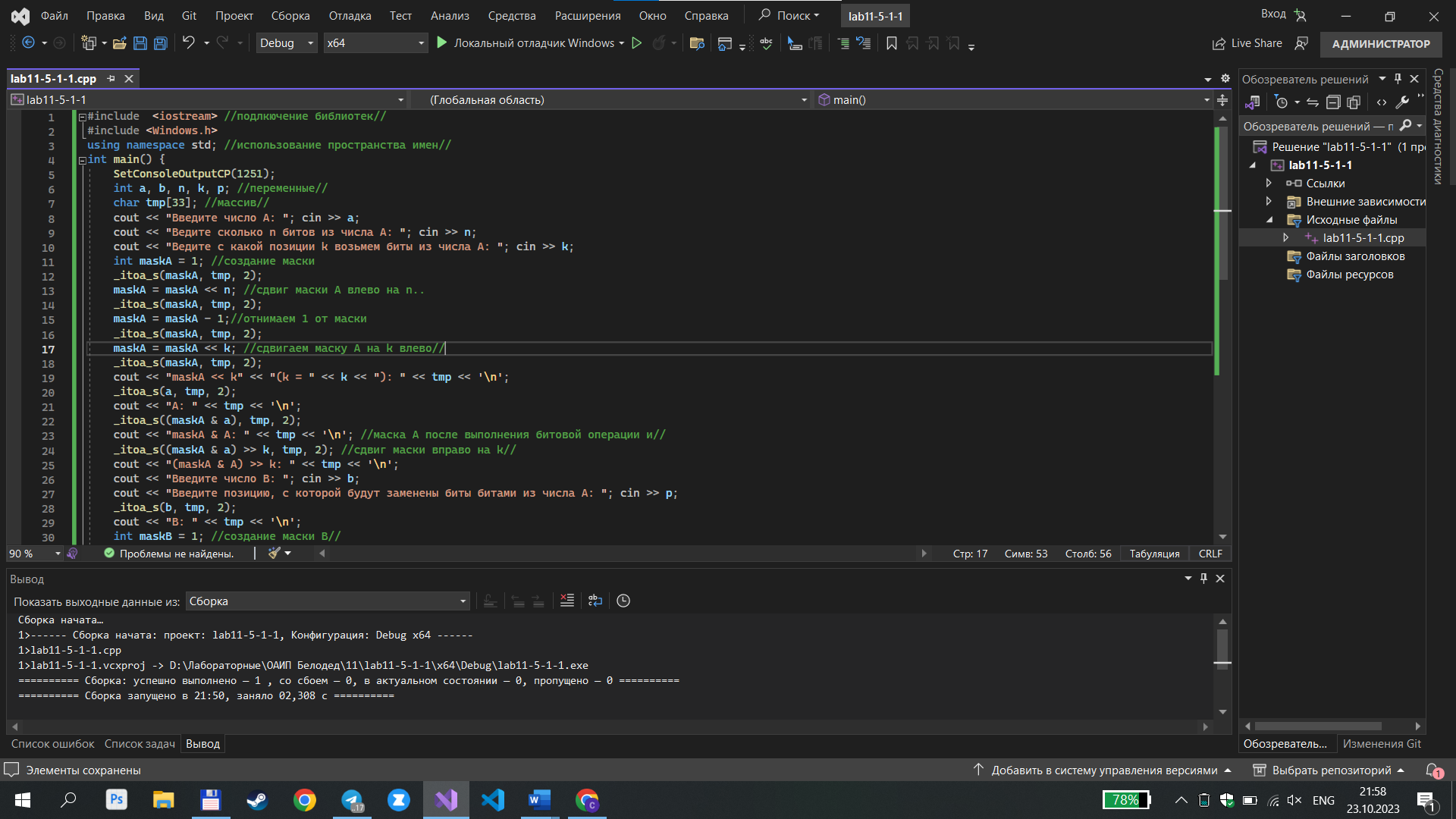
\_itoa\_s(((maskA & a) >> k) << p, tmp, 2); //сдвмг вправо на k//

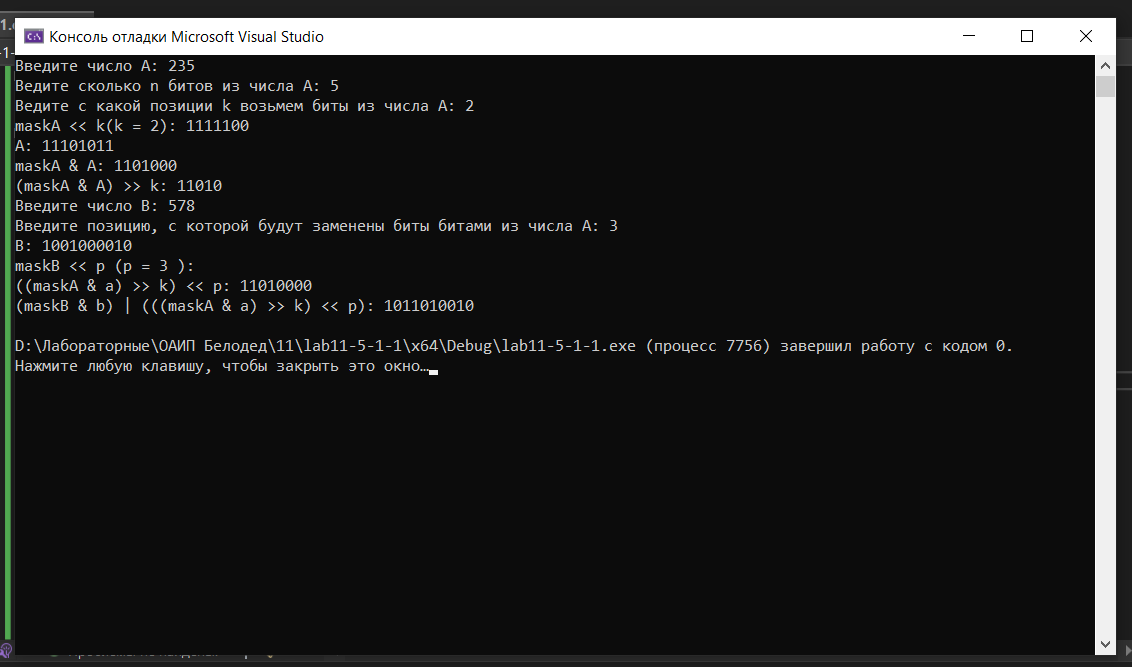
cout << "((maskA & a) >> k) << p: " << tmp << '\n';

\_itoa\_s((maskB & b) | (((maskA & a) >> k) << p), tmp, 2);

cout << "(maskB & b) | (((maskA & a) >> k) << p): " << tmp << '\n'; //вывод конечного результата//

}





2.

#include <iostream> //подключение библиотек//

using namespace std; //использование пространства имен//

int main()

{

setlocale(LC\_CTYPE, "RUS"); //устанавливаем локаль//

int A, p, n, r; //переменные//

char tmp[33];//массив//

cout << "Введите A ";

cin >> A;

\_itoa\_s(A, tmp, 2); //переводим в двоичную систему//

cout << "Представление числа в двоичном коде: " << tmp << endl;

cout << "Введите начальную позицию ";

cin >> p;

cout << "Введите число битов ";

cin >> n;

r = p - n; //вычисляет значение переменной r как разницу между p и n//

while (p>= r)//выполняет цикл, пока значение переменной p больше или равно значению переменной r//

{

A |= 1 << p; //устанавливает бит в позиции p в числе A в единицу, используя побитовую операцию "ИЛИ"//

\_itoa\_s(A, tmp, 2); //преобразует значение переменной A в двоичную строку//

p--; //уменьшает значение переменной p на 1.

}

cout << tmp << endl; //вывод//

}

