Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 11

По дисциплине «Основы алгоритмизации и программирования»

На тему «Одномерные массивы»

Выполнила:

Студентка 1 курса 7 группы

Бабич Виолетта Станиславовна

Преподаватель: асс. Андронова М.В.

2023, Минск

|  |  |
| --- | --- |
| **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>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int A, B;

cout << "Введите число A: ";

cin >> A;

cout << "Введите число B: ";

cin >> B;

//Извлекаем 5 битов из первого числа, начиная со второго

int secBtis = (A >> 1) & ((1 << 5) - 1);

//Очищаем 5 битов в числе В, начиная с третьего бита

int clearMask = ~(((1 << 5) - 1) << 2);

B &= clearMask;

//Вставляем биты из числа А в В, начиная с третьего бита

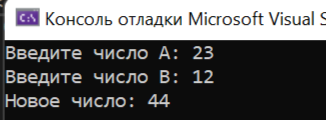
int iBits = secBtis << 2;

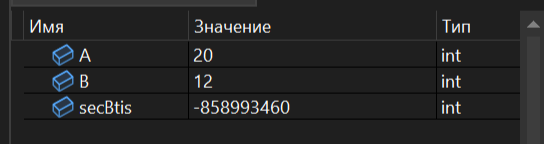
B |= iBits;

cout << "Новое число: " << B << endl;

return 0;

}

****

****

**№2**

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int A, p, n;

cout << "Введите число A: ";

cin >> A;

cout << "Введите позицию р: ";

cin >> p;

cout << "Введите кол-во ббитов n: ";

cin >> n;

//Создаем маску для установки n битов в A, вправо от позиции р

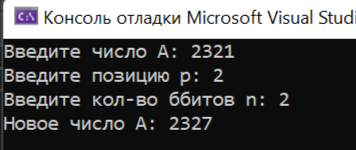
int mask = ((1 << n) - 1) << (p - n + 1);

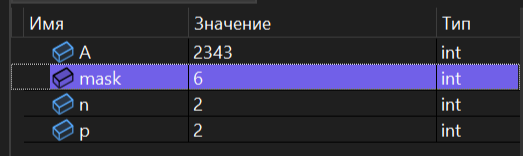
A |= mask;

cout << "Новое число A: " << A << endl;

return 0;

}

****

****

**Дополнительные задания**

|  |  |
| --- | --- |
| **1** | 1. Ввести целое **A** и посчитать, сколько нулей в числе начиная с третьего бита по 13, включая эти биты. 2. Инвертировать в числе **А n** битов вправо от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**. |

**№1**

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int A;

cout << "Введите целое число A: ";

cin >> A;

int mask = 0x1FF8; //Маск, в которой установлен биты с третьего по 13-й

int masked = A & mask;//Применяем маску к числу А

int count = 0;

while (masked > 0) {//Считаем кол-во нулей

if ((masked & 1) == 0) { count++; }

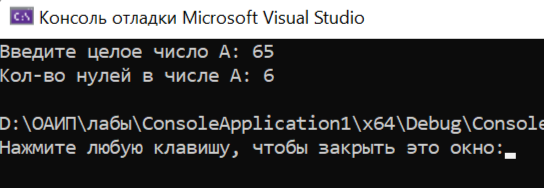
masked >>= 1;

}

cout << "Кол-во нулей в числе А: " << count << endl;

return 0;

}

****

**№2**

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int A = 0b10101010101; //число А

int B = 0b11001100; //число В

int p = 2; //позиция р

int q = 3; //позиция q

int n = 3; //кол-во битов n

int m = 2; //кол-во битов m

//Создаём маску для инвертиврования n битов в А, начиная с р позиции

int maskA = ((1 << n) - 1) << p;

A ^= maskA;

//Создаём маску для выделения m битов из В, начиная с позиции р

int maskB = ((1 << n) - 1) << p;

int bitsB = (B & maskB) >> q;

//Создаём маску для отчистки битов в В

int cleanmaskB = ~maskB;

B &= cleanmaskB;

//Cовмещаем для маски А и В, начиная с позиции q

int inverst = (A >> p) & ((1 << n) - 1);

inverst = ~inverst;

inverst = inverst & ((1 << m) - 1);

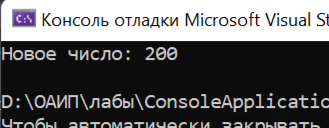
inverst = inverst << q;

B |= inverst;

cout << "Новое число: " << B << endl;

return 0;

}

****

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

**№1**

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int A;

cout << "Введите число А: ";

cin >> A;

//создаем маску для битов со 2-ого по 14-ый

int maskA = ((1 << 13) - 1) << 2;

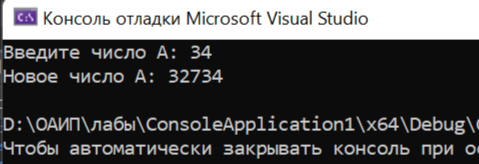
//Применяем маску для интервала битов

A ^= maskA;

cout << "Новое число А: " << A << endl;

return 0;

}

****

**№2**

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int A = 0b10101010101; //число А

int B = 0b11001100; //число В

int p = 2; //позиция р

int q = 3; //позиция q

int n = 3; //кол-во битов n

int m = 2; //кол-во битов m

//Создаём маску для инвертиврования n битов в А, начиная с р позиции

int maskA = ((1 << n) - 1) << p;

A ^= maskA;

//Создаём маску для выделения m битов из В, начиная с позиции q

int maskB = ((1 << n) - 1) << q;

int bitsB = (B & maskB) >> q;

//Создаём маску для отчистки битов в В

int cleanmaskB = ~maskB;

B &= cleanmaskB;

//вставляем n битов из A в B, начиная с позиции q

bitsB = ~bitsB;

bitsB = bitsB & ((1 << m) - 1);

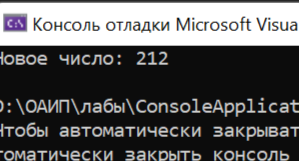
bitsB = bitsB << q;

B |= bitsB;

cout << "Новое число: " << B << endl;

return 0;

}

****

|  |  |
| --- | --- |
| **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>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int A;

cout << "Введите число А: ";

cin >> A;

//проверяем, кратно ли число четырём

if ((A & 3) == 0)

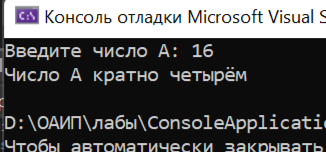
cout << "Число А кратно четырём" << endl;

else

cout << "Число А не кратно четырём" << endl;

return 0;

}

****

**№2**

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int A, p, n;

cout << "Введите число А: ";

cin >> A;

cout << "Введите позицию p: ";

cin >> p;

cout << "Введите кол-во битов n: ";

cin >> n;

//создаем маску для уствновки n битов в A, влево от позиции p

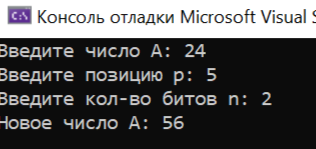
int mask = ((1 << n) - 1) << (p - n + 1);

A |= mask;

cout << "Новое число A: " << A << endl;

return 0;

}

****