Министерство образования и науки Российской Федерации   
Федеральное государственное бюджетное образовательное   
учреждение высшего образования   
«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Астахова Е.В.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

Отчёт

по лабораторной работе № 1

«Логические и побитовые операции»

по дисциплине «Основы программной инженерии»

Студенты группы ПИ 92:

В.М. Шульпов, И.В. Вильк, А. Н. Гулин, Ю.П. Пирязев

Преподаватель доцент, к.п. н. Астахова Е.В.

Барнаул 2019

**Задание 1. Логические битовые операции.**

* 1. **Выполнение установки битов по маске**
* **Код программы**

// lab1\_z1\_1.cpp устанавливает биты по маске (логическая операция И)

#include "stdafx.h"

#include <stdio.h>

#include <locale.h>

#include <Windows.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

system("color F0");

setlocale(LC\_ALL, "Rus");

unsigned int oper, mask, result;//oper - операнд, mask - маска

printf("Программа устанавливает биты по маске\n");

printf("Введите операнд: ");

scanf("%x", &oper);

printf("Введите маску: ");

scanf("%x", &mask);

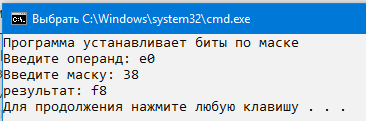
result = oper | mask;

printf("результат: %x\n", result);

return 0;

}

* **Тест программы**



E016=111000002 (операнд)

3816 = 001110002 (маска)

F816 = 111110002 (результат)

* 1. **Выполнение выделения битов по маске**
* **Код программы**

**//** Логические битовые операции

#include "stdafx.h"

#include <Windows.h>

#include <locale.h>

#include <stdio.h>

#include <conio.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "Rus");

system("color F0");

//Выделение битов по маске

int oper2,mask2,result2;

printf("Программа выделяет биты по маске\n");

printf("Введите операнд\n");

scanf("%x",&oper2);

printf("Введите маску\n");

scanf("%x",&mask2);

result2 = oper2 & mask2;

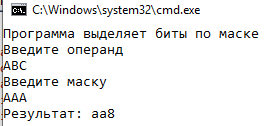
printf("Результат: %x\n",result2);

\_getch();

return 0;

}

* **Тест програмы**

****

ABC16 =1010101111002 - Операнд

AAA 16 =1010101010102 - Маска

AA816 =1010101010002 – Результат

* 1. **Выполнение очистки битов по маске**
* **Код программы**

// OPI.cpp очищает биты по маске (логическая операция &~)

#include "stdafx.h"

#include <stdio.h>

#include <locale.h>

#include <Windows.h>

#include <conio.h>

int main()

{

system("color F0");

setlocale(LC\_ALL, "Rus");

unsigned int oper3, mask3, result3;//oper3 - операнд, mask3 - маска

printf("Программа очищает биты по маске\n");

printf("Введите операнд: ");

scanf("%x", &oper3);

printf("Введите маску: ");

scanf("%x", &mask3);

result3 = oper3 &~ mask3;

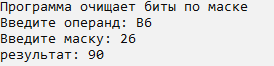
printf("результат: %x\n", result3);

\_getch();

return 0;

}

* **Тест программы**



B616 =101101102 - Операнд

2616 = 1001102 - Маска

9016 =10010002 – Результат

* 1. **Выполнение инвертирования битов по маске**
* **Код программы**

//Лаб.1 №1.4 Инвертирование битов по маске.

#include "stdafx.h"

#include <locale.h>

#include <Windows.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

system("color F0");

setlocale(LC\_ALL, "Rus");

unsigned int a, b; // a - операнд, b - маска.

printf("Инвертирование битов по маске.\n");

printf("Введите операнд а=");

scanf\_s ("%X", &a);

printf("Введите маску b=");

scanf\_s ("%X", &b);

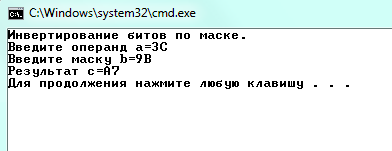
a ^= b; //Инвертирование битов числа a по маске числа b

printf("Результат c=%X\n", a);

return 0;

}

* **Тест программы**



3C16 =001111002 - Операнд

9B 16 =100110112 - Маска

A716 =101001112 – Результат

**Задание 2. Битовые сдвиги.**

* 1. **Выполнение быстрого умножения на 2**
* **Код программы**

//Лаб.1 №2.1 Быстрое умножение на 2 (логический сдвиг влево).

#include "stdafx.h"

#include <locale.h>

#include <Windows.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

system("color F0");

setlocale(LC\_ALL, "Rus");

unsigned int a, b; // a - первый операнд, b - второй операнд.

printf("Логический сдвиг влево.\n");

printf("На число разрядов второго операнда будут сдвинуты разряды первого.\n");

printf("Для быстрого умножения на 2 введите b=1.\n");

printf("Введите первый операнд а=");

scanf\_s ("%X", &a);

printf("Введите второй операнд b=");

scanf\_s ("%X", &b);

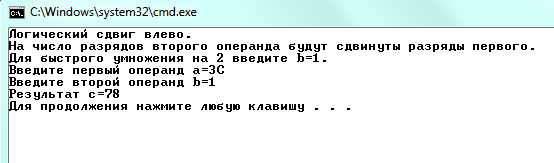
a <<= b; //Логический сдвиг битов операнда "a" влево на число разрядов операнда "b"

printf("Результат c=%X\n", a);

return 0;

}

* **Тест программы**



3C16 =001111002 - Операнд 1

116 =12 - Операнд 2

7816 =011110002 – Результат

* 1. **Выполнение быстрого деления на 2**
* **Код программы**

//Лаб.1 №2.2 Быстрое деление на 2 (логический сдвиг вправо).

#include "stdafx.h"

#include <locale.h>

#include <Windows.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

system("color F0");

setlocale(LC\_ALL, "Rus");

unsigned int a, b; // a - первый операнд, b - второй операнд.

printf("Логический сдвиг вправо.\n");

printf("На число разрядов второго операнда будут сдвинуты разряды первого.\n");

printf("Для быстрого деления на 2 введите b=1.\n");

printf("Введите первый операнд а=");

scanf\_s ("%X", &a);

printf("Введите второй операнд b=");

scanf\_s ("%X", &b);

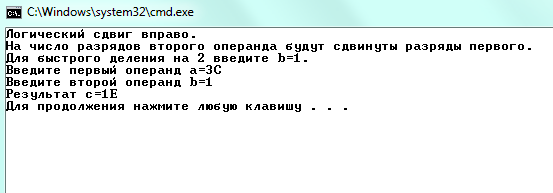
a >>= b; //Логический сдвиг битов операнда "a" вправо на число разрядов операнда "b"

printf("Результат c=%X\n", a);

return 0;

}

* **Тест программы**



3C16 =001111002 - Операнд 1

1 16 =12 - Операнд 2

1E16 =000111102 – Результат

* 1. **Выполнение алгоритма обмена двух переменных через XOR**
* **Код программы**

**#include** "stdafx.h"

#include <stdio.h>

#include <locale.h>

#include <Windows.h>

#include <conio.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

system("color F0");

setlocale(LC\_ALL, "Rus");

int x, y;

printf("Обмен значений двух переменных\n");

printf("Введите x\n");

scanf("%d",&x);

printf("Введите y\n");

scanf("%d",&y);

x = x^y;

y = y^x;

x = x^y;

printf("Значение x теперь равно %d\n",x);

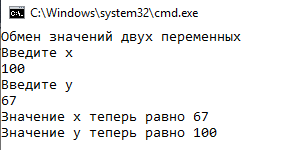
printf("Значение y теперь равно %d\n",y);

\_getch();

return 0;

}

* **Тест программы**

****

* 1. **Реализация циклического сдвига.**
* **Код программы**

// lab1\_z2\_4.cpp выполняет циклический сдвиг

//

#include "stdafx.h"

#include <stdio.h>

#include <locale.h>

#include <Windows.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

system("color F0");

setlocale(LC\_ALL, "Rus");

unsigned long int a; //a- число, которое будем циклически сдвигать

printf("Циклический сдвиг\n");

a = 0xc123456f;

printf("операнд: %x\n", a);

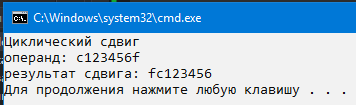
a = a >> 4 | a << 4 \* 7;

printf("результат сдвига: %x\n", a);

return 0;

}

* **Тест программы**



Задача на «4» Вариант 1.Определить разрядность целого числа. Использовать битовый

сдвиг вправо.

* **Код программы**

// lab1\_na4\_1.cpp определяет разрядность числа

//

#include "stdafx.h"

#include <stdio.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

//----- Определение разрядности числа

long int vv = 0x55;

printf("enter number: ");

scanf("%x", &vv);

int i;

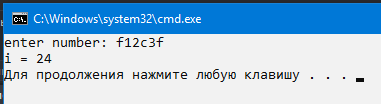
for (i = 0; vv != 0; i++, vv >>= 1);

printf("i = %d\n", i);

return 0;

}

* **Тест программы**



f12c3f16

1111 0001 0010 \*\*\*\* \*\*\*\* \*\*\*\*2

Задача на «5» На факультете биологии АГУ занимаются выведением многоножек. Все многоножки имеют по 32 сегмента. На каждом сегменте слева и/или справа растут ножки. Исследованиями занимаются четыре аспиранта и один студент. Первый аспирант хочет знать, в каких сегментах многоножки есть лапки с обеих сторон. Второй аспирант хочет знать, в каких сегментах многоножки только одна лапка (всё равно с какой стороны). Третий аспирант хочет знать, в каких сегментах есть хотя бы одна лапка Студенту для дипломного проекта нужно знать, есть ли лапка в сегменте с номером N от головы многоножки (справа или слева – он вам укажет).

* **Код программы**

#include "stdafx.h"

#include <stdio.h>

#include <Windows.h>

#include <locale.h>

#include <conio.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

unsigned long long int foot\_l, foot\_r, r\_and\_l, r\_xor\_l, r\_or\_l, n, ost, e, r\_and\_e, l\_and\_e;

char side, right, left;

system("color F0");

setlocale(LC\_ALL, "Rus");

right = 'r';

left = 'l';

e = 0x00000001;//Эта переменна нужна для проверки с помощью операции И

printf("Программа многоножка\n");

printf("Введите сторону, которую запрашивает третий студент(r/l)\n");

scanf("%c",&side);

printf("Введите число соответствующие номеру сегмента, начиная с 0, заканчивая 31\n");

scanf("%d",&n);

printf("Введите код для левой стороны\n");

scanf("%x",&foot\_l);

printf("Введите код для правой стороны\n");

scanf("%x",&foot\_r);

r\_and\_l = foot\_l & foot\_r;//Логическая операция И выбирает только те сегменты, где с двух сторон по ножке(1|1)

r\_xor\_l = foot\_l ^ foot\_r;//Логическая операция Искл.ИЛИ выбирает только те сегменты, где только 1 ножка(0|1,1|0)

r\_or\_l = foot\_l | foot\_r;//Логическая операция ИЛИ выбирает только те сегменты, где хотябы 1 ножка(1|0,1|1,0|1)

printf("Лапки с обоих сторон: %x\n",r\_and\_l);

printf("Лапки только с одной стороны: %x\n",r\_xor\_l);

printf("Лапки хотя бы на одной из сторон:%x\n",r\_or\_l);

if (side == 'r')//Если при запросе стороны написать r, то программа пойдёт по пути правой стороны

{

foot\_r = foot\_r >>n;//Позваляет сдвинуть интересующий студента сегмент в самую правую часть

r\_and\_e = foot\_r & e;//Проверяет самый правый сегмент(после сдвига) на наличие ножки(1/0)

if (r\_and\_e == 1)//1 означает наличие ножки и выводит ответ ДА

{printf("YES\n");}

else//0 означает отсутствие ножки и выводит ответ НЕТ

{printf("NO\n");}

}

else//Если при запросе написать l, то программа пойдёт по пути левой стороны

{

foot\_l = foot\_l >>n;//Позволяет сдвинуть интересующий студента сегмент в самую правую часть

l\_and\_e = foot\_l & e;//Проверяет самый правый сегмент(после сдвига) на наличие ножки(1/0)

if (l\_and\_e == 1)//1 означает наличие ножки и выводит ответ ДА

{printf("YES\n");}

else//0 означает отсутствие ножки и выводит ответ НЕТ

{printf("No\n");}

}

\_getch();

return 0;

}

* **Тест программы**

