## Лабораторная работа 6

## Задача 1

Запустите программу и объясните результат.

## Код программы

#include<stdio.h>

#include<conio.h>

#include <locale.h>

void main()

{

char \*locale = setlocale(LC\_ALL, "");

int i,a,b; // Сколько (битов/байтов) памяти занимают a и b? 2 байта каждая переменная

/\* —————————————————— \*/

printf("\nВведите два целых числа: "); scanf("%d %d",&a,&b);

printf("\n");

/\* ————————————————– \*/

printf("Операция 'инверсия битов' в числе a: 0x%x=%d\n", ~a,~a); /\* Объясните

запись 0x%x=%d. %x - шестнадцатиричное представление числа ~a, %d - целочисленное представление числа ~a \*/

printf("Операция 'инверсия битов' в числе b: 0x%x=%d\n", ~b,~b); /\* Объясните, как

работает операция инверсия. Оператор инверсии меняет ложь на истину, а истину на ложь, для каждого бита \*/

printf("\n");

/\* ————————————————– \*/

printf("Операция 'сдвиг <<2' в числе a : 0x%x=%d\n", a<<2,a<<2); /\* Пропишите в

двоичном коде операцию сдвига на конкретном примере\*/

/\*Например, сдвиг числа 5 влево на 2 позиции 00000101 << 2 == 00010100\*/

printf("Операция 'сдвиг >>2' в числе a : 0x%x=%d\n", a>>2,a>>2); /\* Пропишите в

двоичном коде операцию сдвига на конкретном примере\*/

/\*Например, сдвиг числа 19 вправо на 3 позиции 00010011 >> 3 == 00000010\*/

printf("\n");

/\* ————————————————– \*/

printf("Операция 'битовое И' a&b : 0x%x=%d\n", a&b,a&b); /\* Объясните, как

работает операция битовое И. Побитовое И берёт два числа и логически умножает соответствующие биты.\*/

printf("Операция 'битовое ИЛИ' a|b : 0x%x=%d\n", a|b,a|b); /\* Объясните, как

работает операция битовое ИЛИ. Аналогично работает операция побитового ИЛИ, за исключением того, что она логически суммирует соответствующие биты чисел без переноса.\*/

printf("Операция 'битовое исключающее ИЛИ ': 0x%x=%d\n", a^b,a^b);

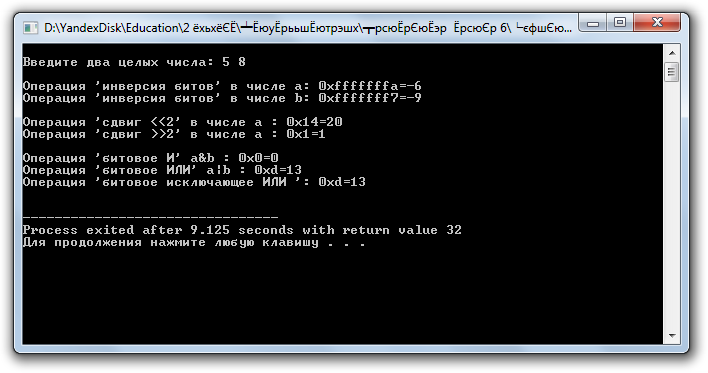
/\* Объясните, как работает операция битовое исключающее ИЛИ. Исключающее ИЛИ применяет побитово операцию XOR. В местах, где биты равны, в результате будут инвентированы, \*/

printf("\n");

getch();

}

## Результат проделанной работы



## Задача 2

Выполните программу и объясните результат. Допишите код.

## Код программы

#include<stdio.h>

#include<conio.h>

#include <locale.h>

void main()

{

char \*locale = setlocale(LC\_ALL, "");

int i,a,b;

/\* —————————————————— \*/

printf("\nВведите два целых числа: "); scanf("%d %d",&a,&b);

printf("\n");

printf("Номера и значения битов старшего байта числа a:\n");

for (i=15;i>=8;i--) /\* Объясните используемый интервал \*/

printf("(%d,%d) ",i,(a>>i)&0x01); /\* Объясните запись (a>>i)&0x01 \*/

printf("\n\n");

/\* —————————————————— \*/

printf("Номера и значения битов младшего байта числа a:\n");

for (i=7;i>=0;i--) /\* Объясните используемый интервал \*/

printf("(%d,%d) ",i,(a>>i)&0x01);

printf("\n\n");

/\* ————————————– \*/

printf("Биты числа b в прямом порядке : ");

for (i=15;i>=0;i--)

printf("%d",(b>>i)&0x01);

printf("\n");

/\* ————————————– \*/

printf("Биты числа b в обратном порядке: "); /\* Допишите код, ссылаясь на

пример: «Биты числа b в прямом порядке» \*/

for (i=0;i<=15;i++)

printf("%d",(b>>i)&0x01);

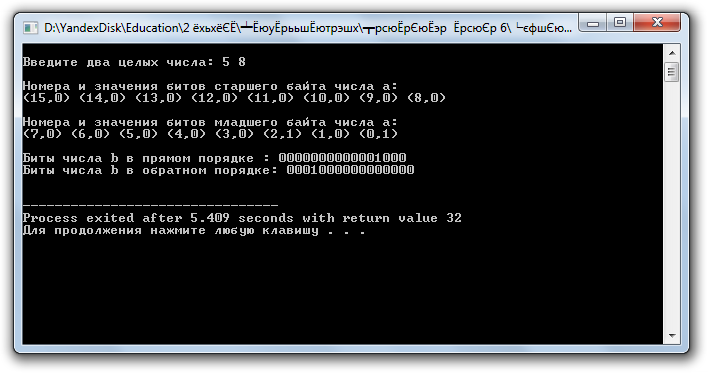
printf("\n\n");

/\* ———————————————— \*/

getch();

}

## Результат выполненной работы



## Задача 3

Объясните, как работает программа.

## Код программы

#include<stdio.h>

#include<conio.h>

#include <locale.h>

void main()

{

char \*locale = setlocale(LC\_ALL, "");

unsigned int n;

printf("Введите натуральное число: "); scanf("%u",&n);

if (n&0x0001)

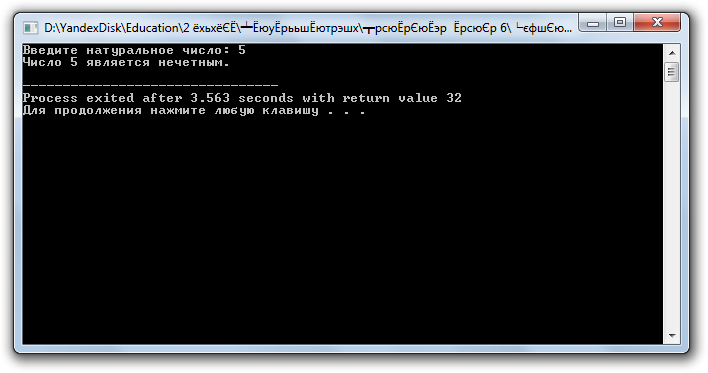
printf("Число %u является нечетным.\n",n);

else printf("Число %u является четным.\n",n);

getch(); /\*Программа получает число и применяет к нему побитовое И, если последний бит в числе будет равен 1, то это число нечетно, если 0, то четно \*/

}

## Результат выполненной работы



## Задача 4

Запустите код и объясните результат программы «Использование операции "сдвиг вправо" для вывода на экран двоичного представления данного натурального числа» Заполните комментарии к программе.

## Код программы

## #include<stdio.h>

## #include<conio.h>

## #include <locale.h>

## void main()

## {

## char \*locale = setlocale(LC\_ALL, "");

## long int a; /\* Исходное натуральное число:

## если a<255, то для хранения числа требуется 4 байта памяти;

## если a<65535, то для хранения числа требуется 4 байта памяти;

## если a<2147483647, то для хранения числа требуется 4 байта памяти \*/

## int n, i; // Параметр цикла

## printf("Введите натуральное число : "); scanf("%lu",&a);

## n=(a>255)?(a>65535)?31:15:7; // Для каждого диапазона значений программа выделяет определнное количество памяти

## printf("Его двоичное представление: ");

## for (i=n;i>=0;i--)

## {

## printf("%d",(a>>i)&0x1); // объясните выражение (a>>i)&0x1

## if (i%8==0) //объясните, для чего служит эта операция. Пробел ставится перед числом в его двоичном представлении, а не после каждой цифры

## printf(" ");

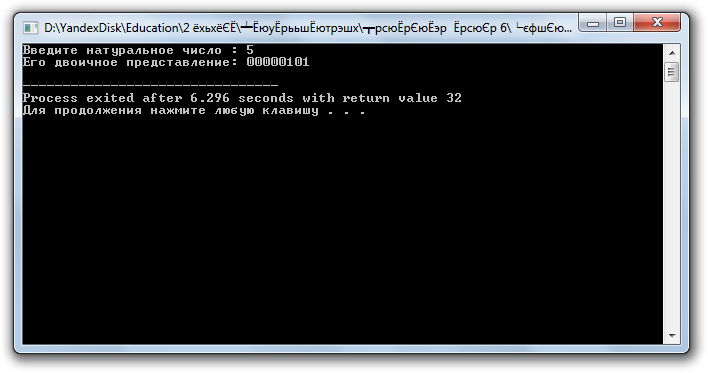
## }

## printf("\n");

## getch();

## }

## Результат выполненной работы



## Задача 5

Выполните программу и объясните результат КАЖДОЙ операции

## Код программы

#include<stdio.h>

#include<conio.h>

void main()

{

int y,x,z,k;

x=03; y=02; z=01;

k=x|y&z; printf("1. %d\n",k); /\* k=(x|(02&01)); (вследствие приоритетов операций) 02&01 -> 102&012 -> 002 k=(x|00); 03|002 -> 112|002 -> 112\*/

k=x|y&~z; printf("2. %d\n",k); /\* Сначала инверсия, потом И, затем ИЛИ \*/

k=x^y&~z; printf("3. %d\n",k); /\* Сначала инверсия, потом И, затем ИЛИ \*/

k=x&y&&z; printf("4. %d\n",k); /\* Сначала И, затем импликация \*/

x=1; y=-1;

k=!x|x; printf("5. %d\n",k); /\* Сначала отрицание, потом ИЛИ \*/

k=-x|x; printf("6. %d\n",k); /\* Операции 6 \*/

k=x^x; printf("7. %d\n",k); /\* Исключающее ИЛИ \*/

x<<=3; printf("8. %d\n",x); /\* Операции 8 \*/

y<<=3; printf("9. %d\n",y); /\* Операции 9 \*/

y>>=3; printf("10. %d\n\n",y); /\* Операции 10 \*/

getch();

}

## Результат выполненной работы

