Лабораторная работа №6

Битовые(поразрядные) операции.

Задача №1:

Постановка задачи:

Запустите программу и объясните результат.

Вопросы заданы в комментариях к коду. Ответы оформите либо в комментариях к коду, либо отдельным блоком после кода

Код:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,a,b; // Сколько (битов/байтов) памяти занимают a и b?
/* ———————— */
printf("\nВведите два целых числа: "); scanf("%d %d",&a,&b);
printf("\n");
/* 32 бита, тк под тип данных int выделяется 32 бита */
printf("Операция 'инверсия битов' в числе a: 0x%x=%d\n", ~a,~a); /* Объясните
запись 0x%x=%d */
printf("Операция 'инверсия битов' в числе b: 0x%x=%d\n", ~b,~b); /* Объясните, как
работает операция инверсия */
printf("\n");
/* Первая запись это представление числа в двоичном виде. Инверсия работает следующим образом: рассматривается двоичная запись числа и все 0 и 1 инвертируются, т.е. вместо 0 становится 1, и вместо 1 0 */
printf("Операция 'сдвиг <<2' в числе a : 0x%x=%d\n", a<<2,a<<2); /* Пропишите в
двоичном коде операцию сдвига на конкретном примере*/
printf("Операция 'сдвиг >>2' в числе a : 0x%x=%d\n", a>>2,a>>2); /* Пропишите в
двоичном коде операцию сдвига на конкретном примере*/
printf("\n");
/* 1) Сдвиг влево:
    10010010 <<2 => 01001000
   2) Сдвиг вправо:
    10010010 >>2 => 00100100

 */
printf("Операция 'битовое и' a&b : 0x%x=%d\n", a&b,a&b); /* Объясните, как
работает операция битовое И */
printf("Операция 'битовое ИЛИ' a|b : 0x%x=%d\n", a|b,a|b); /* Объясните, как
работает операция битовое ИЛИ */
printf("Операция 'битовое исключающее ИЛИ ': 0x%x=%d\n", a^b,a^b);
/* Объясните, как работает операция битовое исключающее ИЛИ */
/*
    1)Операция битового и: Происхрдит побитовое сравнение двух чисел 1 будет только там, где у обоих числе на месте бита стоит 1
    2)Операция битового или: Отличается от битового и, в случае если хотя б у одного числа стоит 1
    3)Операция битового XOR: Возвращает 1 только там где только в одном месте стоит 1.
*/
printf("\n");
getch();
}
```

Задача №2:

Постановка задачи:

Выполните программу и объясните результат.
Допишите код.

Код:

```c
#include<stdio.h>
#include<conio.h>
void main()
{
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");
/* 1) Интервал взят такой потому, что нам нужны биты только старшего байта
   2) Сдвиг вправо на i + коньюнкция с 1. */
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");
/* Интервал связан с тем, что нам необходимы биты младшего байта */
printf("Биты числа b в прямом порядке : ");
for (i=15;i>=0;i-)
printf("%d",(b>>i)&0x01);
printf("\n");
/* Интервал такой потому, что нужны все биты этого числа */
printf("Биты числа b в обратном порядке: "); /* Допишите код, ссылаясь на
пример: «Биты числа b в прямом порядке»
for (;;)
printf("",);
printf("\n\n");*/
/*for (i = 0; i <= 15;i++)
printf("%d",(b>>i)&0x01);
printf("\n\n")*/
getch();
}
```

Вывод:

```
Введите два целых числа: 2
5

Номера и значения битов старшего байта числа a:
(15,0) (14,0) (13,0) (12,0) (11,0) (10,0) (9,0) (8,0)

Номера и значения битов младшего байта числа a:
(7,0) (6,0) (5,0) (4,0) (3,0) (2,0) (1,1) (0,0)

Биты числа b в прямом порядке : 0000000000000101
Биты числа b в обратном порядке:
```

Задача №3:

Постановка задачи:

Объясните, как работает программа.

```c
#include<stdio.h>
#include<conio.h>
void main()
{
  unsigned int n;
  printf("Введите натуральное число: "); scanf("%u",&n);
  if (n&0x0001)
    printf("Число %u является нечётным.\n",n);
  else printf("Число %u является чётным.\n",n);
  getch();
}
```

Код:

```c
#include<stdio.h>
#include<conio.h>
#include<locale.h>
int main()
{
char*locale = setlocale(LC_ALL, "");
unsigned int n;
printf("Введите натуральное число: "); scanf("%u",&n);
if (n&0x0001)
printf("Число %u является неч?тным.\n",n);
else printf("Число %u является ч?тным.\n",n);
getch();
}
/*  Происходит побитовая конъюнкция с единицей и тк в случае возвращение 1 в довичной системе это будет равняться четному числу, тк кратно 2
    то и сумма этих чисел будет четная
  */
```

Вывод:

```
Введите натуральное число: 5
Число 5 является неч?тным.
```

Задача №4:

Постановка задачи:

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

Код:

```c
#include<stdio.h>
#include<conio.h>
#include<locale.h>
int main()
{
char*locale = setlocale(LC_ALL, "");
long int a; /* Исходное натуральное число:
если a<255, то для хранения числа требуется 7 бит памяти памяти;
если a<65535, то для хранения числа требуется 15 бит памяти;
если a<2147483647, то для хранения числа требуется 31 бита памяти */
int i,n; // Параметр цикла
printf("Введите натуральное число : "); scanf("%lu",&a);
n=(a>255)?(a>65535)?31:15:7; // объясните это выражение
printf("Его двоичное представление: ");
for (i=n;i>=0;i--)
{
printf("%d",(a>>i)&0x1); // объясните выражение (a>>i)&0x1
if (i%8==0) //обьясните, для чего служит эта операция
printf(" ");
}
printf("\n");
getch();
}
/* Первое выражение отвечает за то, что в случае попадание числа в поредленный диапазон, происходило выделения нужного кол-ва памяти
    Второе выражение отвечает за вывод всех битов
    Третья операция отвечает за то, чтоб между байтами чисда был пробел, с целью более удобного представления
 */
```

Вывод:

```
Введите натуральное число : 6
Его двоичное представление: 00000110
```