课程名称	软硬件接口程序设计	课程编号	A2130731
实验地点	重庆邮电大学综合实验楼	实验时间	2022.10.27
实验名称	实验四 练习位运算		

一、 实验目的

练习 C 语言中位运算。

二、实验内容

任务 1:

1. 某些数字的右移结果是被定义且可实现的，有些是未定义的。执行给出的包含位运算的程序，并解释结果。

2. 一次右移等效除以二，一次左移等效乘二。执行给出的程序并解释结果。

3. 不能使用位运算符代替逻辑运算符。逻辑运算符（&&，||和！）是 0 或不是 0（通常为 1），但按位运算符可以返回 0 到 2n-1 之间的整数值。执行给出的程序并解释结果。

4. &运算符可用于检查数字是奇数还是偶数。奇数时，表达式（x&1）的值将为非零，否则为零。执行以下程序并解释结果。

任务 2：编写一个程序，计算整数的二进制表达中的位为 1 的个数。程序必须要求用户输入一个以十进制表达的整数，再通过位运算完成计算。

任务 3:

1. 给定一个数组，其中除一个元素外，所有元素出现偶数次，使用 XOR 运算符查找出现奇数次的元素。程序已给出，请解释结果。

2. 使用>>和&运算符查找并显示给定数字的二进制形式。

三、实验过程原始记录(数据、图表等)

任务 1：

1：

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4# ./task1_1
-2 0 -1 0 0 0
```

2：

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4# ./task1_2
x << 1 = 38
x >> 1 = 9
```

3：

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4# ./task1_3
False True root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4#
```

4：

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4#
Oddroot@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab
```

任务 2：

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4# ./task2 48
Your number was 48
In 48, there are 2 bits set to 1
```

任务 3：

1：

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4# vim task3_1.c
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4# gcc task3_1.c -o task3_1
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4# ./task3_1
The odd occurring element is 90 root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab
```

2：

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab4# ./task3_2
Enter Decimal Number : 56
Binary Eqivalent : 0000000000111000root@h9-virtual-machine:/home/h9/ICSI333/Lab/
```

四、实验结果及分析

实验结果可见原始数据记录。相应分析如下：

Task1：

1. -1 左移 1 位，等效乘 2，故结果为-2。1 左移负 1 位，为不合规操作。-1 右移 1 位，右移后补 1，因其补码各位本就为 1，因此右移后仍为 1。1 右移 1 位，为 0。1 左移 32 位，1 左移 64 位，因 int 类型为 32 为，故左溢出，都为 0。

2. 19 的二进制表示形式 10011，左移 1 位为 100110，右移 1 位为 1001。左移时各位 2 的指数都加 1，反之则减 1。

3. &为位运算，各位数值相等则为 True，否则为 False，2 和 5 不等，故为 False。&&为逻辑运算，两端都不为 0 时为 True，2 和 5 都不为 0 故为 True。

4. 因为二进制中各位都是 2 的倍数，故要为基数，则第一位必须为 1。最后一位

为 1 时，其&1 必不为 0，否则为 0。

Task2：通过使用&完成此功能。将被计算的数字不断右移，且每次和 1 进行&操作，计算此时数字的最后一位是否为 1，最终完成对该数所有位的计算。

Task3：

1. ^为异或操作，二者相等时为 0，二者不等时为 1。^运算存在结合率和交换律，且 a^b^b=a。而当数字进行异或时。若两者某位分别为 1 和 0，异或后为 1，若再同 1 异或，则为 0；若再同 0 异或，则为 1，即最后该位显示的为出现基数次的数字的该位。二者分别为 0 和 0，或分别为 1，1 时可同理推导出。再结合交换律性质，故对仅有 1 个数字出现基数次的数组中的所有数进行异或后，最后得到的为出现基数次的数字。

2. mask 初始为 1000 0000 0000 0000，与所给数字进行&操作，可判断其二进制最高位是否为 1，后通过对 mask 不断右移 1 位，依次完成对剩余位数的判断。

五、实验心得体会

1.掌握了 C 语言中的位运算。

2.熟练掌握了左移、右移操作，并且对其原理进行了学习。

3 掌握了位运算和逻辑运算的区别。

4.掌握了异或操作及其基本性质。