第三十五节：移位运算的右移。

【35.1　“右移”运算。】

“右移”运算也是以位为单位进行运算的。位是指二进制中的某一位，位只能是 0 或者 1。欲理解某个数“右移”运算的内部规律，必先把该数展开成二进制的格式，然后才好分析。“右移”运算的符号是“>>”，它的通用格式如下：

```
“保存变量”=“被移数”>>n;
```

运算规律是：“被移数”先被复制一份放到某个隐蔽的临时变量（也称作寄存器），然后对此临时变量展开成二进制的格式，左边是高位，右边是低位，此二进制格式的临时变量被整体由左往右移动了 n 位，原来左边由于数据位移动而新空出的高 n 位数据被直接填入 0，而右边由于数据位移动而导致低 n 位数据被直接覆盖，最后再把移位运算的结果存入“保存变量”。多问一句，这行代码执行完毕后，“保存变量”和“被移数”到底哪个变量发生了变化，哪个变量维持不变？大家记住，只有赋值语句“=”左边的“保存变量”发生数值变化，而右边的“被移数”没有发生变化，因为“被移数”被操作的不是它自己本身，而是它的复制品替身（某个隐蔽的临时变量，也称寄存器）。

上述通用格式中的 n 代表被一次右移的位数，可以取 0，当 n 等于 0 的时候，代表右移 0 位，其实就是数值维持原来的样子没有发生变化。

现在举一个完整的例子来分析“>>”右移运算的规律。有两个 unsigned char 类型的变量 a 和 b，它们的数值都是十进制的 5，求 a=a>>1 和 b=b>>2 的结果分别是多少？分析步骤如下：

第一步：先把 a 和 b 变量原来的数值以二进制的格式展开。十进制转二进制的方法请参考前面第 14,15,16 节的内容。

```
a 变量是十进制 5，它的二进制格式是：  00000101。
b 变量是十进制 5，它的二进制格式是：  00000101。
```

第二步：将 a 右移 1 位，将 b 右移 2 位。

（1）a=a>>1，就是将 a 右移 1 位。

```
a 右移前是        ->    00000101
a 右移 1 位后是   ->    00000010
```

结果分析：把二进制的 00000010 转换成十六进制是：0x02。转换成十进制是 2。所以 a 初始值是 5，右移 1 位后的结果是 2。

（2）b=b>>2，就是将 b 右移 2 位。

```
b 右移前是        ->    00000101
b 右移 2 位后是   ->    00000001
```

结果分析：把二进制的 00000001 转换成十六进制是：0x01。转换成十进制是 1。所以 b 初始值是 5，右移 2 位后的结果是 1。

【35.2　“右移”与除法的关系。】

左移一位相当于乘以 2，而右移跟左移恰恰相反，右移一位相当于除以 2，注意，这里的除法是整除，不带小数点的。比如上面例子，5 右移 1 位就变成了 2（相当于 5 整除 2 等于 2），5 右移 2 位就变成了 1（相当于 5 整除 2 再整除 2 等于 1）。这个现象背后的规律是：在右移运算中，每右移 1 位就相当于整除 2，右移 2 位相当于整除 2 再整除 2，右移 3 位相当于整除 2 再整除 2 再整除 2.....以此类推。这个规律反过来

从除法的角度看，也是成立的：某个数整除 2，就相当于右移 1 位，某个数整除 2 再整除 2 相当于右移 2 位，某个数整除 2 再整除 2 再整除 2 相当于右 3 位.....以此类推。那么问题来了，同样是达到整除 2 的运算结果，从运算速度的角度对比，“右移”和“整除”哪家强？答案是：一条右移语句的运算速度比一条整除语句的运算速度要快很多倍。

【35.3 “右移”的常见应用：不同数据类型之间的分解。】

比如有一个双字节 unsigned int 类型的变量 c,它的初始值是 0x1234,要把它分解成两个 unsigned char 单字节的类型数据 H 和 L,其中 H 是高 8 位字节,L 是低 8 位字节,分解后 H 应该等于 0x12,L 应该等于 0x34,此程序如何写？就需要用到右移。程序分析如下：

```
unsigned char H;        //单字节
unsigned char L;        //单字节
unsigned int c=0x1234; //双字节
L=c;                    //c 的低 8 位直接赋值给单字节的 L
H=c>>8;                 //c 先把高 8 位右移到低 8 位，然后再把这 8 位数据赋值给 H
```

程序运行结果:H 就等于十六进制的 0x12，十进制是 18。L 就等于十六进制的 0x34，十进制是 52.提一个问题，请问执行完上述最后一条语句 H=c>>8 后，此时 c 的值是多少？答案是 c 仍然等于 0x1234，因为 c 本身没有发生变化，只要它没有赋值给它自己，执行完语句后就不会改变它自己本身，也就是本节开篇就提到的：“被移数”被操作的不是它自己本身，而是它的复制品替身（某个隐蔽的临时变量，也称寄存器）。

【35.4 右移运算的“右移简写”。】

当被移数是“保存变量”时，存在“右移简写”。

```
“保存变量”=“保存变量”>>n;
```

上述右移简写如下：

```
“保存变量”>>=n;
```

比如：

```
unsigned char d=8;
unsigned char e=8;

d>>=1; //就相当于 d=d>>1;
e>>=2; //就相当于 e=e>>2;
```

【35.5 例程练习和分析。】

现在编写一个程序来验证刚才讲到的“右移”运算：

程序代码如下：

```
/*---C 语言学习区域的开始。-----------------------------------------------*/

void main() //主函数
{
     unsigned char a=5;
     unsigned char b=5;
```

```
        unsigned char H;              //单字节
        unsigned char L;              //单字节
        unsigned int c=0x1234;        //双字节

        unsigned char d=8;
        unsigned char e=8;

        //右移运算中蕴含着整除 2 的规律。
        a=a>>1;                       //a 右移 1 位，相当于 a=a/2，从原来的 5 变成了 2。
        b=b>>2;                       //b 右移 2 位，相当于 b=b/2/2，从原来的 5 变成了 1。

        //右移的常见应用：不同变量类型的分解。
        L=c;                          //c 的低 8 位直接赋值给单字节的 L
        H=c>>8;                       //c 先把高 8 位右移到低 8 位，然后再把这 8 位数据赋值给 H

        //右移简写。
        d>>=1;                        //就相当于 d=d>>1;
        e>>=2;                        //就相当于 e=e>>2;

        View(a);                      //把第 1 个数 a 发送到电脑端的串口助手软件上观察。
        View(b);                      //把第 2 个数 b 发送到电脑端的串口助手软件上观察。
        View(H);                      //把第 3 个数 H 发送到电脑端的串口助手软件上观察。
        View(L);                      //把第 4 个数 L 发送到电脑端的串口助手软件上观察。
        View(d);                      //把第 5 个数 d 发送到电脑端的串口助手软件上观察。
        View(e);                      //把第 6 个数 e 发送到电脑端的串口助手软件上观察。

        while(1)
        {
        }
    }

    /*---C 语言学习区域的结束。-----------------------------------------------*/
```

在电脑串口助手软件上观察到的程序执行现象如下：

```
开始...

第 1 个数
十进制:2
十六进制:2
二进制:10

第 2 个数
```

```
十进制:1
十六进制:1
二进制:1

第 3 个数
十进制:18
十六进制:12
二进制:10010

第 4 个数
十进制:52
十六进制:34
二进制:110100

第 5 个数
十进制:4
十六进制:4
二进制:100

第 6 个数
十进制:2
十六进制:2
二进制:10
```

分析：

通过实验结果，发现在单片机上的计算结果和我们的分析是一致的。

【35.6　如何在单片机上练习本章节 C 语言程序？】

直接复制前面章节中第十一节的模板程序，练习代码时只需要更改“C 语言学习区域”的代码就可以了，其它部分的代码不要动。编译后，把程序下载进带串口的 51 学习板，通过电脑端的串口助手软件就可以观察到不同的变量数值，详细方法请看第十一节内容。