

第四十节：关系符的等于“==”和不等于“!=”。

【40.1 关系符的等于“==”和不等于“!=”。】

C 语言的“=”并不是等于号，而是赋值的意思，这点前面已讲过。为了跟赋值区分开来，C 语言用“==”来表示等于号的关系符，用“!=”表示不等于的关系符，之所以用“!=”表示不等于的关系，是因为 C 语言中的“!”就是“取非”的运算符，有否定之意。

等于关系符“==”语句的常见格式如下：

```
if(常量或变量==常量或变量)
{
    语句 1;
    语句 2;
}
语句 3;
语句 4;
```

不等于关系符“!=”语句的常见格式如下：

```
if(常量或变量!=常量或变量)
{
    语句 1;
    语句 2;
}
语句 3;
语句 4;
```

上一节讲到，常量或变量在 if 语句中的真假判断，不等于 0 就是真，等于 0 就是假。而本节关系运算符的真假判断也很简单清晰，满足条件就是真，不满足条件就是假。例如：

```
if(2==1)    //2 肯定不等于 1，所以不满足条件“等于的关系”，因此为假，不会执行大括号内的语句。
{
    语句 1;
    语句 2;
}
```

相反，请继续看下面不等于号“!=”这个例子：

```
if(2!=1)    //2 肯定不等于 1，所以满足条件“不等于的关系”，因此为真，会执行大括号内的语句。
{
    语句 1;
    语句 2;
}
```

【40.2 建议把常量放在“==”或“!=”关系符的左边】

“if(a==1)”和“if(1==a)”在实现的功能上是相同的。但是，在实际做项目的时候，还是建议大家采用后面这种写法“if(1==a)”，把常量放在左边，这样写有什么好处？好处是，如果我们不小心把等于号“==”或者“!=”误写成赋值符号“=”时，C编译器在编译时，它能及时发现错误并且报错告知我们，因为常量在左边是无法赋值的，编译器能及时发现错误。但是如果常量在右边而变量在左边，因为变量是允许赋值的，所以有一些C语言编译器未必会报错，就会留下不易察觉的程序隐患。比如：

```
if(a==5)
{
    语句 1;
}

if(b!=2)
{
    语句 2;
}
```

建议改成：

```
if(5==a)
{
    语句 1;
}

if(2!=b)
{
    语句 2;
}
```

【40.3 例程练习和分析。】

现在编写一个实验程序，一共有8个给定的数，要统计其中数值“等于85”的数有几个，统计其中数值“不等于75”的数有几个。

程序代码如下：

```
/*---C 语言学习区域的开始。-----*/

unsigned char x1=90; //给定的第 1 个数
unsigned char x2=65; //给定的第 2 个数
unsigned char x3=85; //给定的第 3 个数
unsigned char x4=79; //给定的第 4 个数
unsigned char x5=95; //给定的第 5 个数
unsigned char x6=65; //给定的第 6 个数
unsigned char x7=75; //给定的第 7 个数
unsigned char x8=85; //给定的第 8 个数
```

```
unsigned char a=0; //统计等于 85 的变量总数
unsigned char b=0; //统计不等于 75 的变量总数

void main() //主函数
{
    //第一部分:统计“等于 85”的总数有多少个。
    if(85==x1) //把常量 85 放在等于号的左边
    {
        a++; //相当于 a=a+1, 用来统计等于 85 的总数
    }

    if(85==x2) //把常量 85 放在等于号的左边
    {
        a++; //相当于 a=a+1, 用来统计等于 85 的总数
    }

    if(85==x3) //把常量 85 放在等于号的左边
    {
        a++; //相当于 a=a+1, 用来统计等于 85 的总数
    }

    if(85==x4) //把常量 85 放在等于号的左边
    {
        a++; //相当于 a=a+1, 用来统计等于 85 的总数
    }

    if(85==x5) //把常量 85 放在等于号的左边
    {
        a++; //相当于 a=a+1, 用来统计等于 85 的总数
    }

    if(85==x6) //把常量 85 放在等于号的左边
    {
        a++; //相当于 a=a+1, 用来统计等于 85 的总数
    }

    if(85==x7) //把常量 85 放在等于号的左边
    {
        a++; //相当于 a=a+1, 用来统计等于 85 的总数
    }

    if(85==x8) //把常量 85 放在等于号的左边
    {
        a++; //相当于 a=a+1, 用来统计等于 85 的总数
    }
}
```

```
}
```

```
//第二部分:统计“不等于 75”的总数有多少个。
```

```
if(75!=x1) //把常量 75 放在不等于号的左边
```

```
{
```

```
    b++; //相当于 b=b+1, 用来统计不等于 75 的总数
```

```
}
```

```
if(75!=x2) //把常量 75 放在不等于号的左边
```

```
{
```

```
    b++; //相当于 b=b+1, 用来统计不等于 75 的总数
```

```
}
```

```
if(75!=x3) //把常量 75 放在不等于号的左边
```

```
{
```

```
    b++; //相当于 b=b+1, 用来统计不等于 75 的总数
```

```
}
```

```
if(75!=x4) //把常量 75 放在不等于号的左边
```

```
{
```

```
    b++; //相当于 b=b+1, 用来统计不等于 75 的总数
```

```
}
```

```
if(75!=x5) //把常量 75 放在不等于号的左边
```

```
{
```

```
    b++; //相当于 b=b+1, 用来统计不等于 75 的总数
```

```
}
```

```
if(75!=x6) //把常量 75 放在不等于号的左边
```

```
{
```

```
    b++; //相当于 b=b+1, 用来统计不等于 75 的总数
```

```
}
```

```
if(75!=x7) //把常量 75 放在不等于号的左边
```

```
{
```

```
    b++; //相当于 b=b+1, 用来统计不等于 75 的总数
```

```
}
```

```
if(75!=x8) //把常量 75 放在不等于号的左边
```

```
{
```

```
    b++; //相当于 b=b+1, 用来统计不等于 75 的总数
```

```
}
```

```
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。

    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:7

十六进制:7

二进制:111

分析：

变量 a 为 2。（等于 85 的有 x3, x8 这 2 个）

变量 b 为 7。（不等于 75 的有 x1, x2, x3, x4, x5, x6, x8 这 7 个）

通过实验结果，发现在单片机上的实验结果和我们的分析是一致的。

【40.4 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。