

第四十四节：小括号改变判断优先级。

【44.1 小括号的概述。】

小括号在 C 语言里一直有“强制、改变优先级、明确顺序”这层含义。C 语言中，凡是在判断语句里插入了小括号，程序就会优先执行最里面小括号的判断语句，之后才会根据判断符的优先级执行其它相关语句。此功能很实用，因为 C 语言的判断符号众多，非常不利于程序员记忆各种不同符号的优先级顺序，而小括号却解决了这个问题。只要在合适的地方插入恰当的小括号，就可以强制改变判断的优先级，有了此项功能就不用程序员再刻意去记忆繁杂的优先级，同时，也为实际项目带来两个好处，一个是明确判断顺序，另一个是改变判断顺序。多说一句，哪怕添加的小括号是多余的重复的啰嗦的，也不会对程序带来副作用，反而，只会给程序员内心带来更大的确定和安全感。比如：

两个 if 条件判断语句：

```
if(a>=70&&a<=80) 和 if(a<70 || a>80)
```

有一些朋友喜欢插入两个小括号变成：

```
if((a>=70)&&(a<=80)) 和 if((a<70) || (a>80))
```

在这里插入的小括号是多余的重复的啰嗦的，但是还好，不会对程序有副作用。上述的修改，在不知道“>、>=、<、<=”这类语句跟“&&、||”这类语句哪个优先级更高的前提下，插入了小括号，可以更加明确判断的顺序，这种做法也值得肯定。

【44.2 小括号的具体应用。】

我个人平时在面对同时存在“>、>=、<、<=”和“&&、||”这些语句时，由于我很清楚“>、>=、<、<=”比“&&、||”这类语句的优先级更高，所以我不需要在此插入小括号来明确判断的顺序。但是遇到下面这种情况，我是一定会通过插入小括号的方式来明确判断的顺序。什么情况呢？如下：

```
if(“判断条件 1” || “判断条件 2 ” && “判断条件 3”)
```

这种情况下，就会很容易让我出现一个疑问，到底是先“判断条件 1”跟“判断条件 2”相“或”，最后再跟“判断条件 3”相“与”？还是先“判断条件 2”跟“判断条件 3”相“与”，最后再跟“判断条件 1”相“或”？如果此时果断插入小括号，就可以很容易明确它们的先后顺序，减少内心不必要的纠结。

插入小括号的第 1 种情况：

```
if((“判断条件 1” || “判断条件 2 ”) && “判断条件 3”)
```

插入小括号的第 2 种情况：

```
if(“判断条件 1” || (“判断条件 2 ” && “判断条件 3”))
```

上述两种情况，具体选择哪一种判断顺序要根据项目的需要来决定。同样的 3 个“判断条件”，如果插入的小括号的位置不一样，判断的顺序就不一样，那么结果也可能出现不一样，比如，上述判断条件：

假设“判断条件 1”为“真”，

假设“判断条件 2”为“真”，

假设“判断条件 3”为“假”，

等效成如下：

插入小括号的第 1 种情况：

```
if( (真||真) &&假)
{
    语句 1;
}
```

这种情况下，先判断最里面小括号的真假，（真||真）的结果是“真”，然后再把结果“真”和外面的“假”进行“与”判断，（真&&假）的结果是“假”，所以上述的最终判断是“假”，不能执行“语句 1”。

插入小括号的第 2 种情况：

```
if(真|| (真&&假))
{
    语句 1;
}
```

这种情况下，先判断最里面小括号的真假，（真&&假）的结果是“假”，然后再把结果“假”和外面的“真”进行“或”判断，（真||假）的结果是“真”，所以上述的最终判断是“真”，能执行“语句 1”。

综合上述两种情况，对比之后，得出这样的结论：在同样的条件和关系下，如果插入不同位置的小括号，就可以得出不同的结果。也就是说，小括号可以让关系判断变得丰富起来，可以实现各种复杂的逻辑判断功能。

【44.3 例程练习和分析。】

现在编写一个实验程序验证上述两种判断顺序。

程序代码如下：

```
/*---C 语言学习区域的开始。-----*/

//x, y 这三个变量作为条件判断的变量
unsigned char x=5;
unsigned char y=6;

//a, b 这两个变量作为输出判断结果的真假，0 代表假，1 代表真。
unsigned char a=0; //默认为 0，也就是默认为假
unsigned char b=0; //默认为 0，也就是默认为假

void main() //主函数
{
    if((x<y||y>x)&&x==y) //里面的条件是((真||真)&&假)，最终结果判断是假
    {
        a=1;
    }

    if(x<y|| (y>x&&x==y)) //里面的条件是(真||(真&&假))，最终结果判断是真
```

```

    {
        b=1;
    }

    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:0

十六进制:0

二进制:0

第 2 个数

十进制:1

十六进制:1

二进制:1

分析：

变量 a 为 0。（0 代表此条件判断结果为假）

变量 b 为 1。（1 代表此条件判断结果为真）

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。

【44.4 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。