

第二十一节：减法运算的 5 种常见格式。

【21.1 减法语法格式。】

减法语法格式：

“保存变量” = “减数 1” - “减数 2” - ... - “减数 N”；

含义：右边的“减数”与“减数”相减（这里暂时把平时所说的被减数也归类为减数），并且把最终的运算结果赋值给左边的“保存变量”。注意，这里的符号“=”不是等于号的意思，而是赋值的意思。左边的“保存变量”必须是变量，不能是常量，否则编译时会报错。右边的“减数”既可以是变量，也可以是常量，也可以是“保存变量”本身自己。多说一句，什么是变量和常量？变量是在程序中被更改的，被分配的一个 RAM 空间。常量往往是数字，或者被分配在 ROM 空间的一个具体数值。下面根据右边“减数”与“减数”的不同组合，列出了减法运算的 5 种常见格式。

第 1 种：“减数 1”是常量，“减数 2”是常量。比如：

```
unsigned char a;  
a=15-3;
```

分析：数字“15”和“3”都是常量。执行上述语句后，保存变量 a 变成了 12。

第 2 种：“减数 1”是变量，“减数 2”是常量。比如：

```
unsigned char b;  
unsigned char x=15;  
b=x-10;
```

分析：x 是变量，“10”是常量。由于原来 x 变量里面的数值是 15，执行上述语句后，保存变量 b 变成了 5。而变量 x 则保持不变，x 还是 15。

第 3 种：“减数 1”是变量，“减数 2”是变量。比如：

```
unsigned char c;  
unsigned char x=15;  
unsigned char y=6;  
c=x-y;
```

分析：x 是变量，y 也是变量。由于原来 x 变量里面的数值是 15，y 变量里面的数值是 6，执行上述语句后，保存变量 c 变成了 9。而变量 x 和 y 则保持不变，x 还是 15，y 还是 6。

第 4 种：“减数 1”是保存变量本身，“减数 2”是常量。比如：

```
unsigned char d=18;  
d=d-2;  
d=d-7;
```

分析：d 是保存变量，“2”和“7”都是常量。这类语句有一个特点，具备了自减功能，可以更改自己本身的数值。比如原来保存变量 d 的数值是 18，执行“d=d-2;”语句后，d 变成了 16，接着再执行完“d=d-7;”语句后，d 最后变成了 9。

第 5 种：“减数 1”是保存变量本身，“减数 2”是变量。比如：

```
unsigned char e=28;  
unsigned char x=15;
```

```
unsigned char y=6;
e=e-x;
e=e-y;
```

分析：e 是保存变量，x 与 y 都是变量。这类语句有一个特点，具备了自减功能，可以更改自己本身的数值。比如原来保存变量 e 的数值是 28，执行“e=e-x;”语句后，e 变成了 13，接着再执行完“e=e-y;”语句后，e 最后变成了 7。

【21.2 例程练习和分析。】

现在我们编写一个程序来验证上面讲到的 5 个减法例子：

程序代码如下：

```
/*---C 语言学习区域的开始。-----*/

void main() //主函数
{
    unsigned char a;    //定义一个变量 a，并且分配了 1 个字节的 RAM 空间。
    unsigned char b;    //定义一个变量 b，并且分配了 1 个字节的 RAM 空间。
    unsigned char c;    //定义一个变量 c，并且分配了 1 个字节的 RAM 空间。
    unsigned char d=18; //定义一个变量 d，并且分配了 1 个字节的 RAM 空间。初始化默认为 18.
    unsigned char e=28; //定义一个变量 e，并且分配了 1 个字节的 RAM 空间。初始化默认为 28.

    unsigned char x=15; //定义一个变量 x，并且分配了 1 个字节的 RAM 空间。初始化默认为 15.
    unsigned char y=6;  //定义一个变量 y，并且分配了 1 个字节的 RAM 空间。初始化默认为 6.

    //第 1 种：“减数 1”是常量，“减数 2”是常量。
    a=15-3;

    //第 2 种：“减数 1”是变量，“减数 2”是常量。
    b=x-10;

    //第 3 种：“减数 1”是变量，“减数 2”是变量。
    c=x-y;

    //第 4 种：“减数 1”是保存变量本身，“减数 2”是常量。
    d=d-2;
    d=d-7;

    //第 5 种：“减数 1”是保存变量本身，“减数 2”是变量。
    e=e-x;
    e=e-y;

    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(c); //把第 3 个数 c 发送到电脑端的串口助手软件上观察。
View(d); //把第 4 个数 d 发送到电脑端的串口助手软件上观察。
View(e); //把第 5 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:12
十六进制:C
二进制:1100

第 2 个数
十进制:5
十六进制:5
二进制:101

第 3 个数
十进制:9
十六进制:9
二进制:1001

第 4 个数
十进制:9
十六进制:9
二进制:1001

第 5 个数
十进制:7
十六进制:7
二进制:111

```

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。

【21.3 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。