



普通高等教育“十三五”规划教材
基于移动终端学习的创新教材



C language

C 语言程序设计

C YUYAN CHENGXU SHEJI

主编 杨禹军

 吉林大学出版社

任务 5.1 while 循环语句——数据统计

任务引入

当数据表中有多条记录时，如何统计和分析表中的相关数据？本任务主要统计表 2-2 中 3~7 月份的餐饮费和通信费的总和。利用 while 循环结构即可完成统计。所谓循环结构是按照一定的条件，控制重复执行某个程序段的一种结构。



理论知识

1. while 循环语句的一般格式

由 while 语句构成的循环也称当循环，while 循环的一般格式如下：

while (表达体)

{循环体}

例如：

while ($x \leq 0$) $x++$;

说明：

①while 是 C 语言的关键字。

②while 后面的一对圆括号中的表达式用来控制循环体是否执行，一般为关系表达式或逻辑表达式，也可以是 C 语言其他类型的合法表达式。

③后面循环体是循环重复执行的部分，可以是基本语句、控制语句，也可以是用花括号括起来的复合语句。

2. while 循环语句的执行过程

while 循环语句的执行过程如图 5-1 所示。

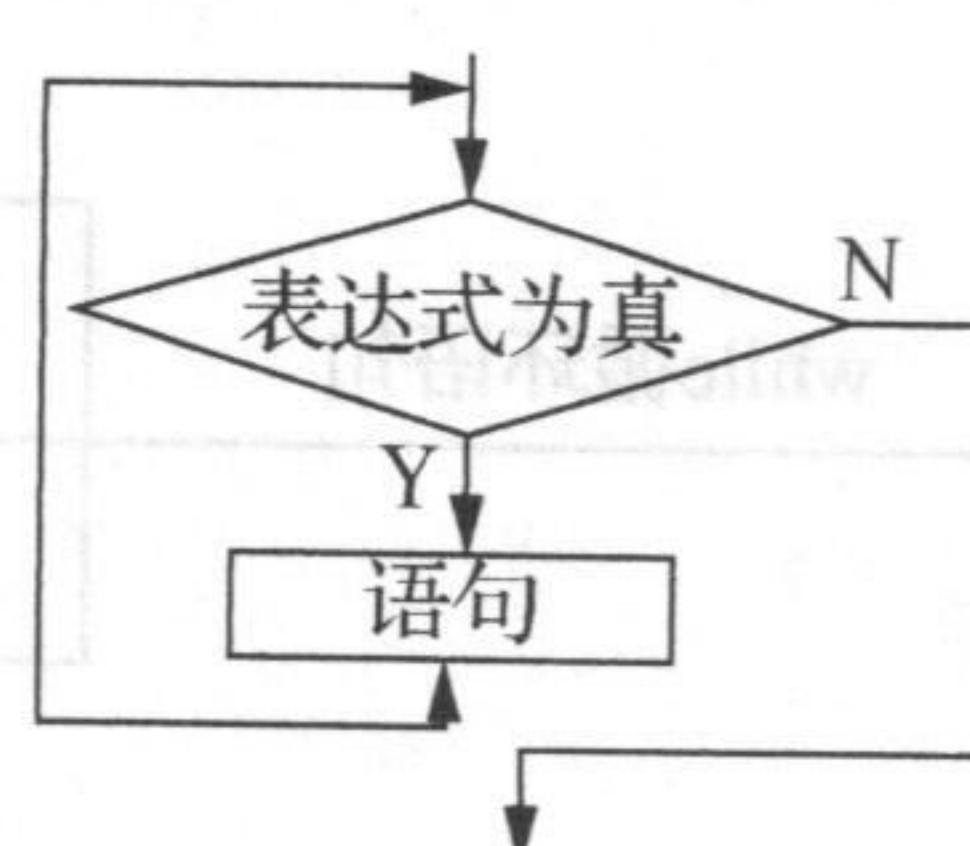


图 5-1 while 循环语句的执行过程

①计算 while 后一对圆括号中的表达式的值。当值为非 0 时，执行步骤②；当值为 0 时，执行步骤④。

②执行循环体中的语句。

③转去执行步骤①。

④退出 while 循环。

注意：while 后一对圆括号中的表达式的值决定了循环体是否执行，因此，进入 while 循环后，一定要有能使表达式的值变为 0 的操作，否则，循环将会无限制地进行下去。

有如下程序段：

```
int x=4;
while (x>=0) x--;
```

while 循环的作用是：当 x 的值大于或等于 0 时，执行 x-- 语句，直到 x 的值小于 0 时结束循环。

设整型变量 a 的当前值为 0，有如下循环语句：

```
while (a++<=2)
    printf ("%d\n", a);
```

while 的循环体是 printf 语句，每执行一次循环体输出一次 a 的值；循环判断条件 ($a++ \leq 2$) 表示取 a 的值与 2 比较，之后 a 自增；只要 a 自增前的值小于或等于 2，就执行循环体，直到 a 等于 3 时，表达式 $a++ \leq 2$ 的值是 0，之后 a 自增为 4 退出循环。执行该语句的输出结果是分行输出 1、2、3。

如果将上面程序段变成如下形式：

```
while (a++<=2);
    printf ("%d\n", a);
```

此时循环体就不再是 printf 语句，而是一条空语句。此时程序段的输出与不加分号时不同，而是在退出循环后，输出 a 的当前值 4。

【例 5.1】 编写程序，求 $1+2+3+\cdots+100$ 的值。

思路：这是一个求 100 个数累加和的问题。所加的加数从 1 变化到 100，可以看到加数是有规律变化的：后一个数比前一个数增 1，第一个加数为 1，最后一个加数为 100。因此可以在循环中使用一个整型变量 i，每循环一次 i 的值增 1，一直循环到 i 的值超过 100。但是要注意 i 的初值应是 1。

第二个要考虑的问题是如何求累加和。首先应该定义一个变量 sum 存放这 100 个数的累加和，并设 sum 的初值为 0。第一次求 $sum = sum + i$ 时，就是将 $0+1$ 的和存放在 sum 中。然后把 sum 中的值加上 2 再存放在 sum 中，以此类推，直到循环终止得到 100 个数的和。

以下是求累加和的典型算法。

```
#include<stdio.h>
void main ()
{
    int i, sum;
    i=1;
    sum=0; /* sum 的初值为 0 */
    while (i<=100) /* 循环体中累加一次，i 增 1 */
    {
        sum=sum+i;
```

```

    语句 i++；代码告；⑤表达式语句，即 0 非表达式，值由表达式中字符圆括一值 ofidw 值长①
    }
    printf (" sum=%d\n", sum);
}

```

程序运行后的输出结果如下：

sum=5050

任务实施

本任务是输入表中所列的数据即 3、4、5、6、7 月份的生活费，计算这 5 个月份的生活费之和。

```

void main ()
{
    int i=3, shf, sum=0;
    while (i<=7)
    {
        printf (" 请输入%d月的生活费:", i);
        scanf ("%d", &shf);
        sum=sum+shf;
        i++;
    }
    printf (" 3~7 月份生活费合计:%d\n", sum);
}

```

运行结果如图 5-2 所示。

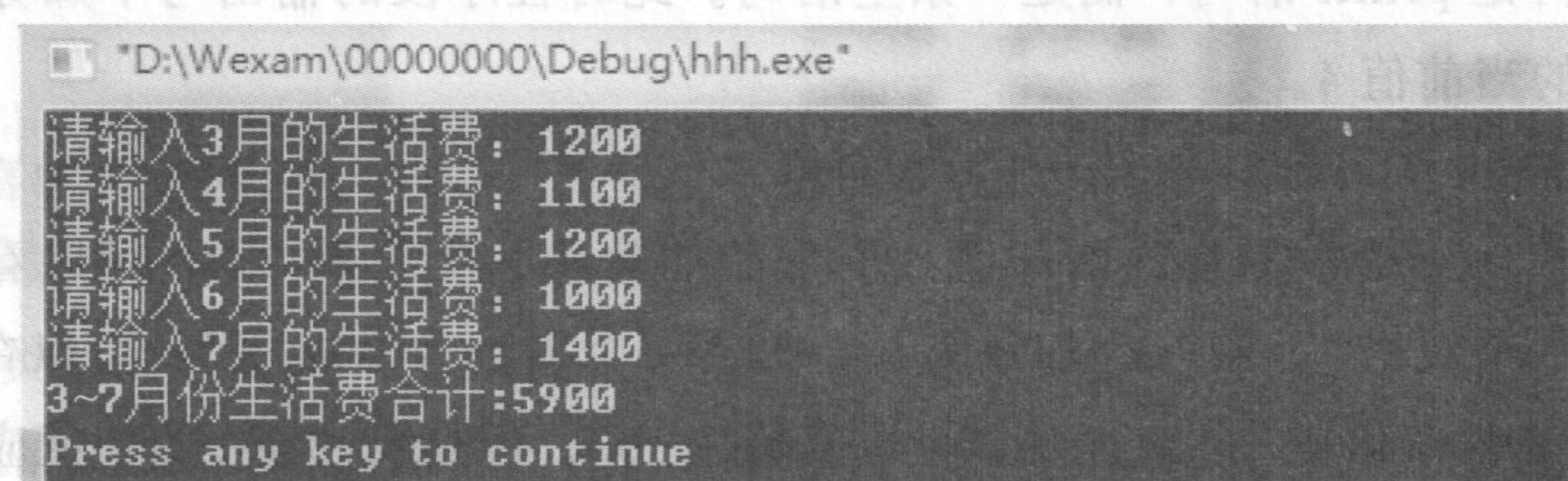


图 5-2 运行结果

任务 5.2 do while 循环语句——数据统计

任务引入

利用 do-while 循环语句，统计 3~7 月份兼职获得的收入总和。



理论知识

5.2.1 do-while 循环语句

1. do-while 循环语句的一般格式

do-while 循环语句又称直到型循环语句，它的一般格式如下：

```
do
{循环体}
while (表达式);
```

例如：

```
do x=2
    while (x>5);
```

说明：

- ① do 是 C 语言的关键字，必须和 while 联合使用。
- ② do-while 循环由 do 开始，以 while 结束。必须注意的是：在 while (表达式) 后面的 “;” 不能丢，它表示 do-while 语句的结束。
- ③ while 后面的圆括号中的表达式用于进行条件判断，决定循环体是否执行。
- ④ do 后面的循环体可以是一条可执行的语句，也可以是由多条语句构成的复合语句，对于复合语句一定要用花括号将其括起来。

2. do-while 循环语句的执行过程

do-while 循环语句的执行过程如图 5-3 所示。

- ① 执行循环语句。
- ② 求表达式的值。当值非 0 时，转去执行步骤①；当值为 0 时，执行步骤③。
- ③ 退出 do-while 循环。

注意：和 while 语句一样，在 do-while 循环语句中，一定要有能使表达式的值变为 0 的操作；否则，循环将会无限制地进行下去。

while 循环和 do-while 循环重要的区别是：while 循环的控制表达式出现在循环体之前，只有当表达式的值为“真”时，才会执行循环体，因此，如果表达式的值一开始就为“假”，则循环体一次也不执行；do-while 循环中是先执行循环体，然后才判断表达式的值，所以，无论一开始表达式的值为“真”或“假”，循环体至少执行一次。

【例 5.2】 用 do-while 循环语句计算 1~100 所有整数之和。

设计思路与用 while 循环语句实现的思路是一样的，同样需要一个变量 i，每循环一次使 i 的值增 1，一直循环到 i 的值超过 100，并且 i 的初值为 1；需要一个变量 sum 存放这 100 个数的累加和，sum 的初值为 0。具体不同之处见以下程序段：

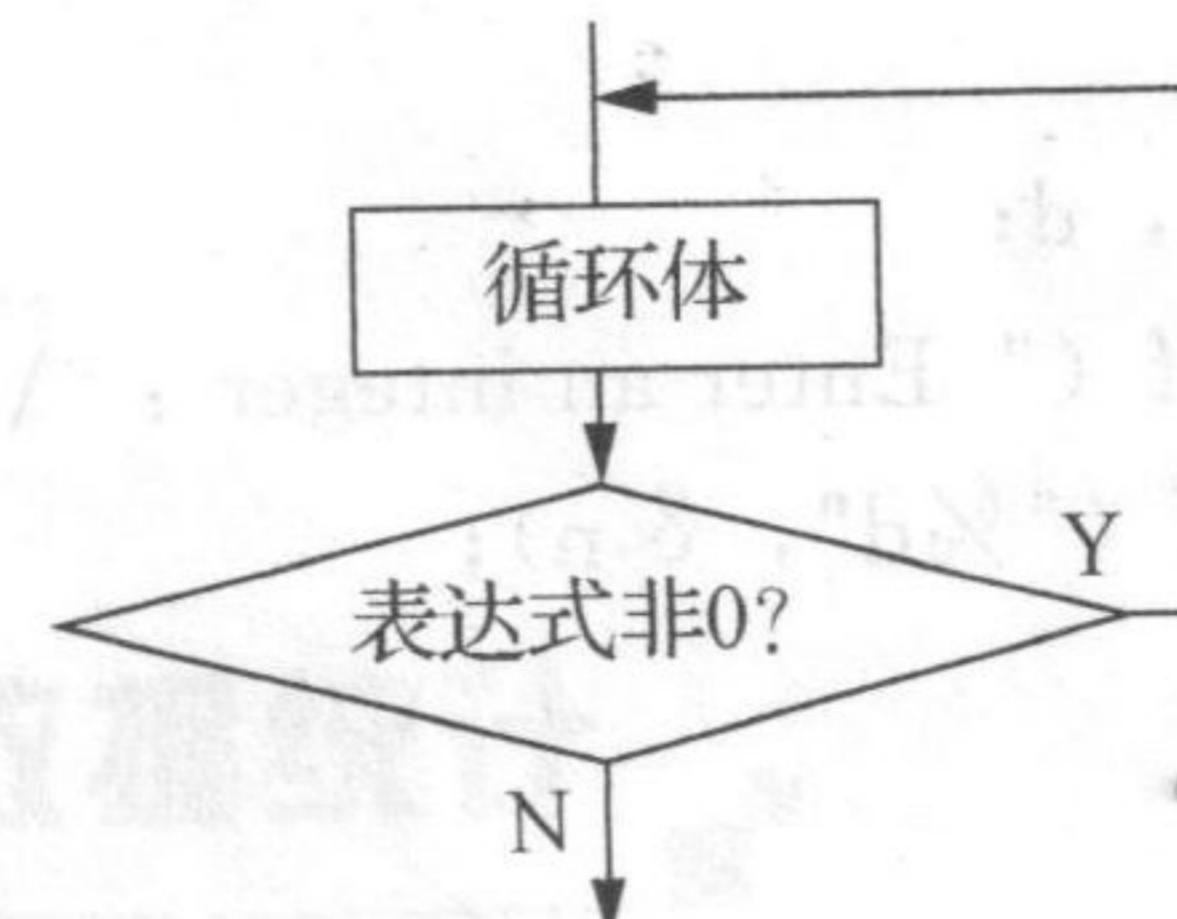


图 5-3 do-while 循环语句的执行过程

```
#include<stdio.h>
void main ()
{
    int i, sum;
    i=1;
    sum=0;           /* sum 的初值为 0 */
    do
    {
        sum=sum+i;
        i++;          /* 循环体中累加一次, i 增 1 */
    } while (i<=100);
    printf (" sum=%d\n", sum);
}
```

程序运行后输出结果如下：

```
sum=5050
```

从上面程序段可以看出，用 do-while 循环实现该运算与使用 while 循环的实现相比，唯一区别就是执行循环体和判断表达式值的顺序不同，但是两种循环得到的结果是一样的。

【例 5.3】 从键盘上输入一个整数，然后把这个整数的各位逆序输出。例如，输入 456，输出 654。

思路：所谓的逆序输出，就是先输出整数的个位，再输出十位，以此类推。可以通过除 10 取余的方法获得任意整数的个位数字。例如：当 $n=456$ 时， $456 \% 10 = 6$ 。然后用 $n/10$ 可以将 n 缩小到 $1/10$ 倍，如 $456/10=45$ ，可以看到原来的三位数变成了两位数。以此类推就可以将整数的各位数字按逆序依次输出。程序如下：

```
#include<stdio.h>
void main ()
{
    int n, d;
    printf (" Enter an integer : \n" );
    scanf ("%d", &n);
    do
    {
        d=n%10;
        printf ("%d", d);
        n/=10;
    } while (n!=0);
    printf ("\n" );
}
```

程序的运行情况如下：

```
Enter an integer: 456↙
654
```



任务实施

```
#include <stdio.h>
void main ()
{
    int i=3, jzsr, sum=0;
    do
    {
        printf ("请输入%d月的兼职收入:", i);
        scanf ("%d", &jzsr);
        sum=sum+jzsr;
        i++;
    } while (i<=7);
    printf ("3~7月份兼职收入总和:%d\n", sum);
}
```

运行结果如图 5-4 所示。

```
D:\Wexam\00000000\Debug\hhh.exe
请输入3月的兼职收入: 200
请输入4月的兼职收入: 500
请输入5月的兼职收入: 460
请输入6月的兼职收入: 800
请输入7月的兼职收入: 120
3~7月份兼职收入总和:2080
Press any key to continue.
```

图 5-4 运行结果