

3.1 相关知识点

1. if 语句

if 语句的三种形式：

(1) if 语句基本形式,如图 3.1 所示。

```
if(表达式)  
    语句;
```

(2) if-else 形式,如图 3.2 所示。

```
if(表达式)  
    语句 1;  
else  
    语句 2;
```

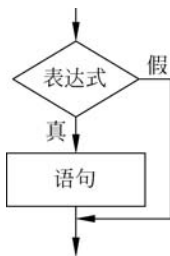


图 3.1 if 语句基本形式

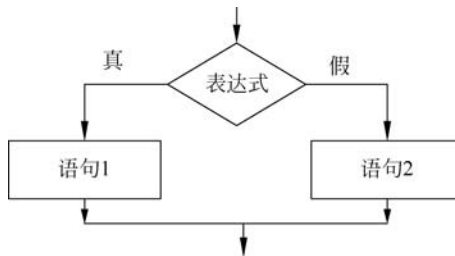


图 3.2 if-else 形式

(3) if 语句嵌套形式,如图 3.3 所示。

```
if(表达式 1)        语句 1;  
else if(表达式 2)    语句 2;  
...  
else if(表达式 n)    语句 n;  
else                 语句 n + 1;
```

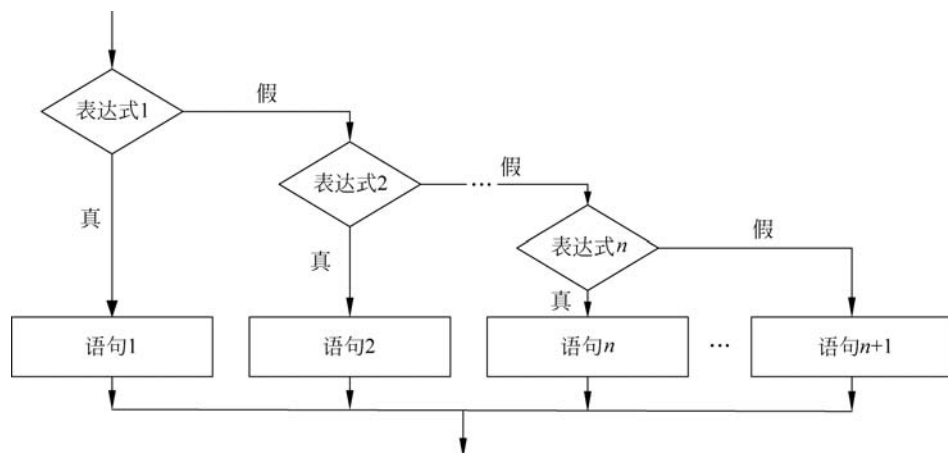


图 3.3 if 语句嵌套形式

if 语句在使用过程中需要注意的地方：

(1) 在以上三种形式的 if 语句中,if 后面的括号中均为表达式。该表达式通常是逻辑表达式或者关系表达式,但也可以是其他类型表达式,如赋值表达式等,甚至也可以是一个变量。例如:if(x=2)和if(x)都是允许的。只要表达式或变量的值为非 0,即为“真”。只有当表达式或变量的值为 0 时,才为“假”。比如在 if(x=2)中,由于赋值表达式 x=2 的值永远都为非 0,所以无论变量 x 的值为多少,该表达式均为“真”。在计算表达式的值时,特别注意区分关系运算符“==”和赋值运算符“=”的区别,要判断两个值是否相等,使用“==”,而不是“=”。

(2) 注意存在以下等价关系:if(x)等价于 if(x!=0); if(!x)等价于 if(x==0)。

(3) 在 if 语句中,条件判断表达式必须用圆括号括起来,在语句之后必须加分号。

(4) 在 if 语句的三种形式中,所有的语句均应为单个语句,如果想在满足一定条件时执行一组语句,则需要把这一组语句用{}括起来,使其构成一条复合语句。

(5) 当 if 语句中的执行语句又是 if 语句时,则构成了 if 语句的嵌套形式。在嵌套内的 if 语句可能又是 if-else 形式的,这将会出现多个 if 和多个 else 重叠的情况,这时特别要注意 if 和 else 的配对问题。if 和 else 配对的原则为:else 总是和位于它上面的、离它最近的、未配对的 if 进行配对,以上三个条件缺一不可。

2. switch 语句

switch 语句的一般形式,如图 3.4 所示。

```

switch(表达式)
{
    case 常量表达式 1:[语句序列 1];break;
    case 常量表达式 2:[语句序列 2]; break;
    ...
    case 常量表达式 n:[语句序列 n]; break;
    [default:语句序列 n+1; break; ]
}
  
```

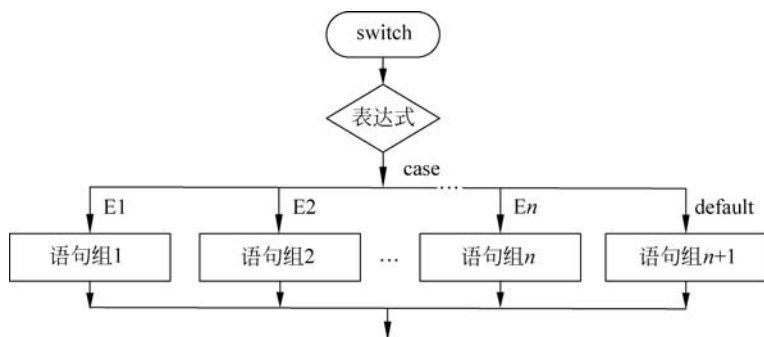


图 3.4 switch 语句

switch 语句在使用过程中需要注意的地方：

(1) switch 语句的执行过程。计算 switch 关键字后面表达式的值,并逐个与 case 后面的常量表达式的值进行比较,当表达式的值与某个常量表达式的值相等时,即执行其后的语句。如果表达式的值与所有 case 后面的常量表达式的值均不相等时,则执行 default 后面的语句。

(2) switch 后面的表达式必须是整型表达式或字符表达式。

(3) case 后面跟的表达式必须是整型或字符型的常量表达式。

(4) 所有 case 后的常量表达式的值不能相同,否则会出现错误。

(5) 在 case 后面可以是多条语句,而且可以不用{}括起来。

(6) 可以没有 default 分支。

(7) 注意 break 关键字的使用。break 用于 case 分支语句组的后面,表示程序执行到此时,该 break 所在的 switch 语句结束,不再执行后面的内容。如果一个 case 分支中没有 break,则在执行完该分支后,仍然会继续无条件地执行下面分支中的语句。例如,在下列 switch 语句中,如果变量 x 的值为 1,由于没有在 case 1 分支后面加入 break,所以程序在执行完 case 1 分支后仍然会执行 case 2 分支,因此最终输出结果为“x==1 x== 2”,而不是“x==1”。

```

switch(x)
{
    case 1: printf("x== 1 ");
    case 2: printf("x== 2 ");
}

```

3.2 实验目的

(1) 熟练掌握各种选择结构,包括 if-else 及其嵌套结构,if-else if-else 形式的多重选择结构,以及 switch 形式的多重选择结构的使用。

(2) 熟练掌握选择结构中测试表达式的书写。

(3) 学会用选择结构编制简单的程序。

3.3 实验内容

3.3.1 程序设计

1. 编制程序

要求：任给三个整数 a 、 b 、 c ，将最大数存放在变量 a 中，最小数存放在变量 c 中，并按从大到小的顺序输出。

【指导】

这是一个简单的数据排序的问题，其主要操作是比较和交换，算法步骤如下：

- (1) 比较 a 和 b ，如果 $a < b$ ，则将 a 、 b 的值进行交换。
- (2) 比较 a 和 c ，如果 $a < c$ ，则将 a 、 c 的值进行交换。
- (3) 比较 b 和 c ，如果 $b < c$ ，则将 b 、 c 的值进行交换。
- (4) 按顺序输出 a 、 b 、 c 。

【流程图】

流程图见图 3.5。

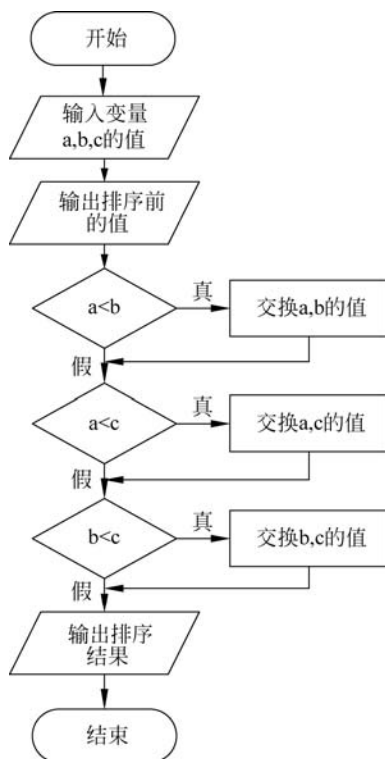


图 3.5 流程图

【参考程序】

```
#include <stdio.h>
int main()
{
```

```

int a, b, c, temp;
printf("Input a,b and c:\n");
scanf("%d%d%d", &a, &b, &c);
printf("Before sorting: a = %d, b = %d, c = %d\n", a, b, c);
if (a < b) { temp = a; a = b; b = temp; }
if (a < c) { temp = a; a = c; c = temp; }
if (b < c) { temp = b; b = c; c = temp; }
printf("After sorting: a = %d, b = %d, c = %d\n", a, b, c);
return 0;
}

```

2. 调试程序示例

邮局对邮寄包裹的收费标准为：每件收手续费 0.2 元，不同重量的邮资按表 3.1 所示进行计算。

表 3.1 邮寄不同重量包裹收费标准

重量/kg	收费标准/元
小于 10	0.80
大于或等于 10 但不超过 20	0.75
大于或等于 20	0.70

编写程序，输入包裹的重量，输出所需的邮资（输出结果保留两位小数）。

【流程图】

流程图见图 3.6 所示。

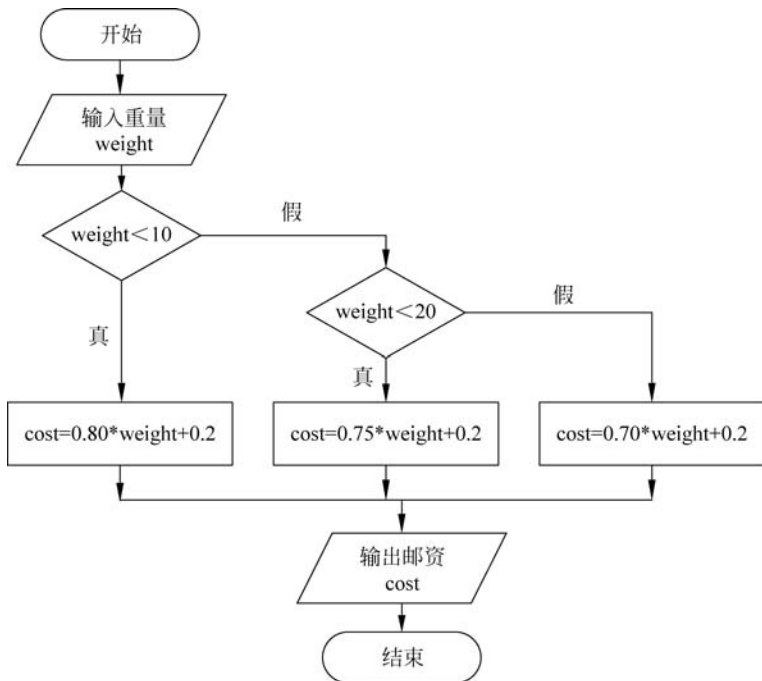


图 3.6 流程图

【参考程序】

```
#include <stdio.h>
void main()
{
    float weight, cost;
    printf("Enter weight: \n");
    scanf("%f", &weight);
    if(weight < 10)
        cost = 0.80 * weight + 0.2;
    else if(weight < 20)
        cost = 0.75 * weight + 0.2;
    else
        cost = 0.70 * weight + 0.2;
    printf("Delivery cost is %.2f\n", cost);
}
```

【说明】

(1) 单击如图 3.7 所示的“添加或移除按钮”菜单中的“自定义”命令,弹出如图 3.8 所示的“自定义”对话框,选中“工具栏”下的“调试”选项,主界面将显示调试工具栏(如图 3.9 所示);或者此步骤直接单击主菜单上的“调试”按钮,会显示出调试的各个功能步骤。

(2) 程序调试开始,单击调试工具栏中的按钮 (逐语句(F11))。该按钮的功能是单步执行,即单击一次执行一行(如图 3.10 所示),编辑窗口中的箭头指向某一行,表示程序将要执行该行。图 3.10 中下方是局部变量窗口和监视窗口,在监视窗口中可以改变变量的值。

(3) 单击按钮 (逐语句(F11))两次,程序执行到输入语句这一行(如图 3.11 所示),同时运行窗口(图 3.12)显示 Enter weight:,继续单击按钮 (逐语句(F11)),在运行窗口中输入 25(图 3.13),按 Enter 键后,箭头指向了 if(weight<10)这一行(如图 3.14 所示),在局部变量窗口中可以看到变量 weight 的值是 25.000000。

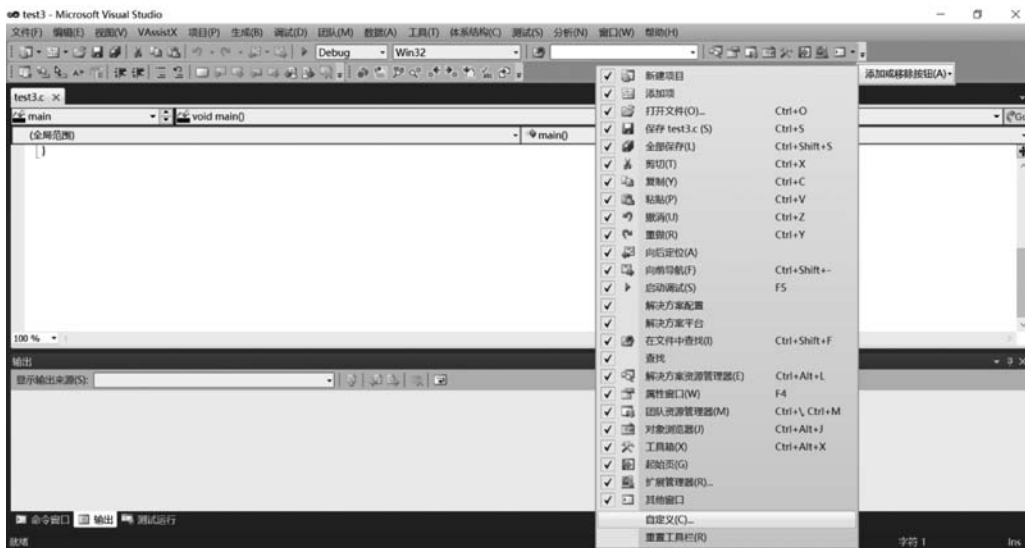


图 3.7 选择工具栏



图 3.8 “自定义”对话框

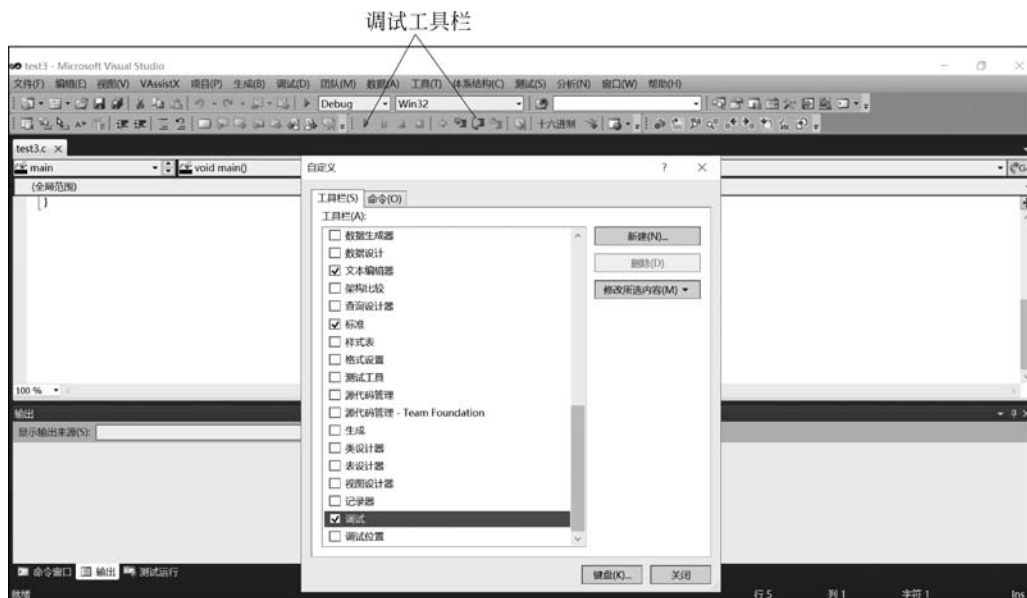
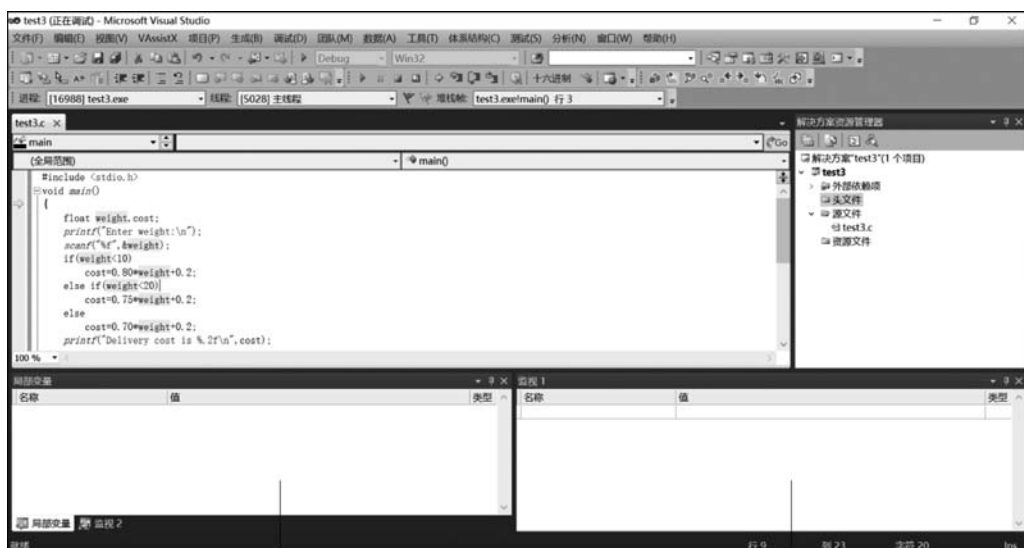


图 3.9 调试工具栏



局部变量窗口

监视窗口

图 3.10 程序调试开始

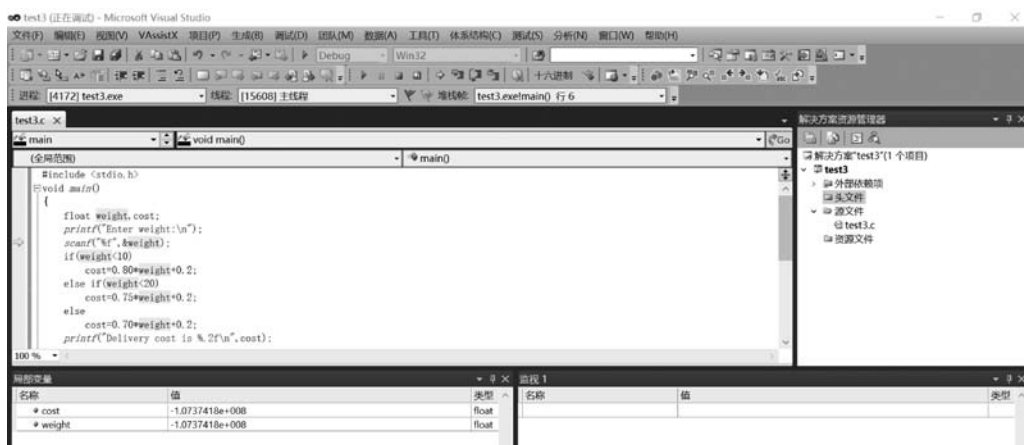


图 3.11 程序单步调试

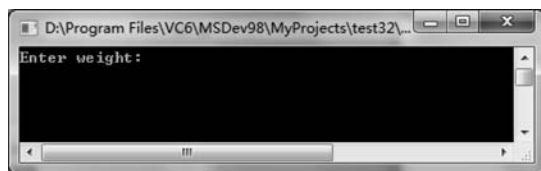


图 3.12 运行窗口

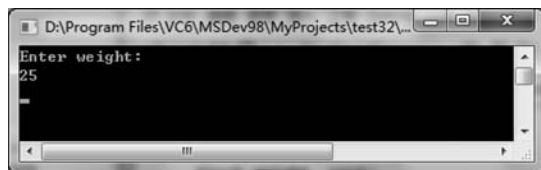


图 3.13 在运行窗口中输入变量 weight 的值

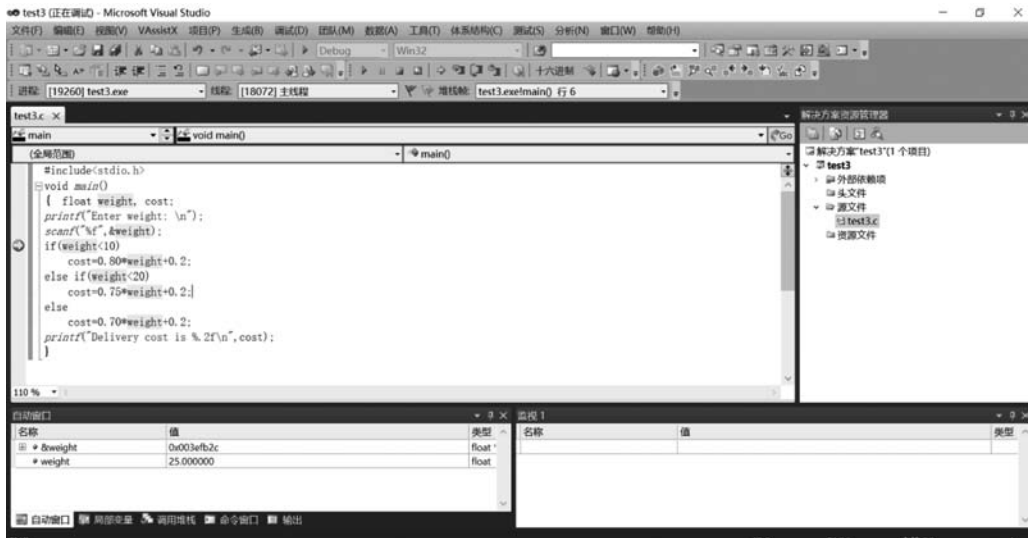


图 3.14 程序单步调试,显示变量 weight 的值

(4) 继续单击按钮  (Step Over) 三次,箭头指向 printf 这一行(如图 3.15 所示),在变量窗口可以看到变量 cost 的值为 17.700001。

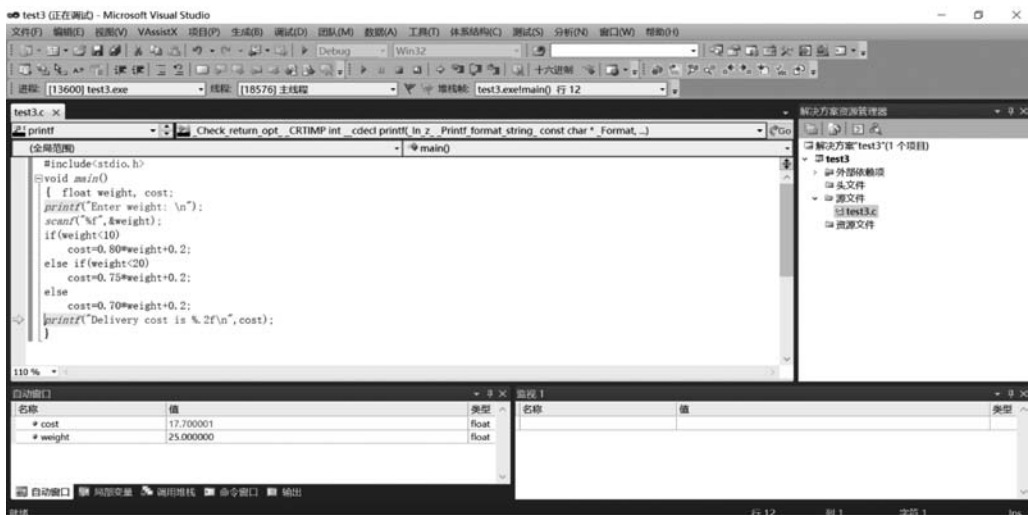


图 3.15 程序单步调试

(5) 继续单击按钮  (Step Over),运行窗口显示运行结果(如图 3.16 所示)。

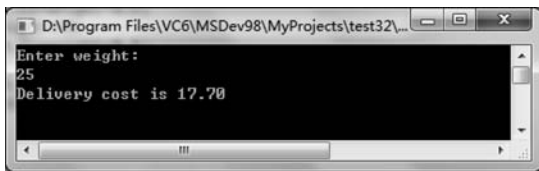


图 3.16 在运行窗口中显示结果

(6) 单击按钮  (Stop Debugging), 程序调试结束。

3. 编制程序

该程序的功能是根据考试分数划分成绩等级。成绩等级按如下方法分级: 分数大于或等于 90 分的为 A 级, 大于或等于 80 且小于 90 分则为 B 级, 大于或等于 70 且小于 80 分则为 C 级, 大于或等于 60 且小于 70 分为 D 级, 如果分数小于 60 则为 E 级。假设考试分数均为整数, 要求用 switch 语句实现。

【指导】

算法分析: 首先, 定义一个整型变量 score, 保存考试分数; 然后, 用 score 除以 10, 结果为一个介于 0~10 的整数, 再用这个结果作为 switch 语句的表达式, 与不同的 case 分支进行匹配。

【参考程序】

```
#include <stdio.h>
int main()
{   int score;                //存放考试分数的变量
    int n;
    char grade;               //存放成绩等级
    scanf("%d", &score);      //输入考试成绩
    n = score/10;
    switch (n)                 //判定成绩等级
    {   case 10:
        case 9: grade = 'A'; break;
        case 8: grade = 'B'; break;
        case 7: grade = 'C'; break;
        case 6: grade = 'D'; break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0: grade = 'E'; break;
        default: printf("Input error!\n"); return; //输入的分数不在 0~100
    }
    printf("Grade is %c\n", grade); //输出成绩等级
    return 0;
}
```

3.3.2 程序分析

运行下面的程序, 分析运行结果。

```
#include <stdio.h>
int main()
{   int x = 1, y = 0, a = 0, b = 0;
    switch(x)
    {   case 1:
        switch(y)
        {   case 0:  a++;  break;
```

```

        case 1:  b++;  break;
    }
    case 2:  a++;b++; break;
    case 3:  a++;b++;
}
printf("\na = %d,b= %d",a,b);
return 0;
}

```

【指导】

(1) 输入程序源代码,编译、连接后运行程序,显示程序的结果为: a=2,b=1。

(2) 对程序运行结果的分析: 这是一个 switch 语句的嵌套形式。外层 switch 中, x 的值为 1, 所以进入 case 1 分支中, 即执行内层 switch 语句。在内层 switch 中, y 的值为 0, 进入内层 switch 的 case 0 分支, 此时执行 a++, a 的值变为 1, 然后执行 break 语句。由于 break 只能终止并跳出最近一层的 switch 语句, 所以此处的 break 是跳出了内层的 switch 语句。在外层 switch 语句中, case 1 分支中并没有 break 语句, 因此会继续执行 case 2 分支中的语句组, 执行了 a++ 和 b++ 后, a 的值变为了 2, b 的值变为 1, 在此处遇到了 break 语句终止并跳出了外层的 switch 语句。因此, 最后输出 a 和 b 的值分别为 2 和 1。

3.4 思考题

(1) 企业发放的奖金根据利润提成, 用 switch 语句实现。利润 I 低于(含等于)100 000 元的, 奖金可提 10%; 利润高于 100 000 元, 低于(含等于)200 000 元时, 低于 100 000 元的部分按 10% 提成, 高于 100 000 元的部分, 可提成 7.5%; 当 200 000 元 $< I \leq 400 000$ 元时, 低于 200 000 元的部分仍按上述办法提成(下同), 高于 200 000 元的部分按 5% 提成; 当 400 000 元 $< I \leq 600 000$ 元时, 高于 400 000 元的部分按 3% 提成; 当 600 000 元 $< I \leq 1 000 000$ 元时, 高于 600 000 元的部分按 1.5% 提成; 当 $I > 1 000 000$ 元时, 超过 1 000 000 元的部分按 1% 提成。从键盘输入当月利润 I, 求应发奖金总数。

(2) 输入三角形三条边 a, b, c 的值, 判断这三条边能否构成三角形。若能, 还要显示该三角形是等边三角形、等腰三角形、直角三角形或任意三角形。

提示: 判断构成三角形的条件是任意两边之和大于第三边。

(3) 编写程序计算存款的本金合计。已知银行整存整取存款不同期限的利率分别为: 半年 2.55%, 一年 2.75%, 二年 3.35%, 三年 4.00%, 五年 4.75%。要求输入存钱的本金和期限, 输出到期时的本金和利息合计。

提示: 利用输入函数分别输入本金和存款年限。使用 switch 语句处理不同期限的情况, 根据本金及相应的利率算出利息。注意, switch 语句中的表达式类型必须是整型或字符型, 对于半年(0.5 年)的情况, 应在 switch 语句外单独处理。

(4) 编写程序计算飞机票款。输入舱位代码和购票数量, 输出总票款。

提示: 国内客票的舱位等级主要分为头等舱(舱位代码为 F)、公务舱(舱位代码为 C)、经济舱(舱位代码为 Y); 经济舱里面又分不同的座位等级(舱位代码为 B、H、K、L、M、N、Q、T、X 等, 价格也不一样)。票价规则为: F 舱为头等舱公布价, C 舱为公务舱公布价, Y 舱

为普通舱(经济舱)公布价,B 舱为普通舱 9 折,H 舱为普通舱 8.5 折,K 舱为普通舱 8 折,L 舱为普通舱 7.5 折,M 舱为普通舱 7 折,N 舱为普通舱 6.5 折,Q 舱为普通舱 6 折,T 舱为普通舱 5.5 折,X 舱为普通舱 5 折。程序首先输入 F 舱、C 舱和 Y 舱的公布价,然后输入舱位代码和购票数量,利用 switch 语句处理不同折扣的情况,计算出机票款并输出。输入、输出都要有文字说明,结果取 2 位小数。

(5) 编写程序,计算并输出分段函数的值。

$$y = \begin{cases} x^2 + x - 6, & x < 0 \\ x^2 - 5x, & 0 \leq x < 5 \\ x^3 - 1, & \text{其他} \end{cases}$$