

3.1 相关知识

1. if 语句

if 语句的三种形式：

(1) if 语句基本形式(参见图 3.1)如下：

```
if(表达式)语句
```

(2) if...else 形式(参见图 3.2)如下：

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

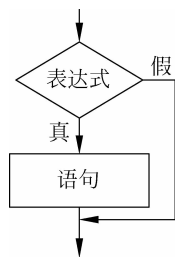


图 3.1 if 语句基本形式

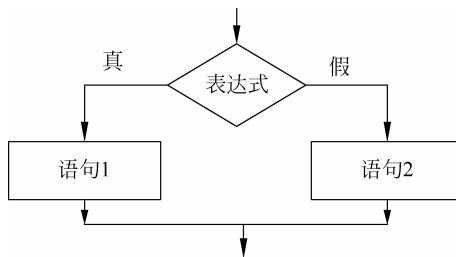


图 3.2 if...else 形式

(3) if 语句嵌套形式(图 3.3)如下：

```
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;
```

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的表达式,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”。例如在 if(x=2)中,由于赋值表达式 x=2 的值

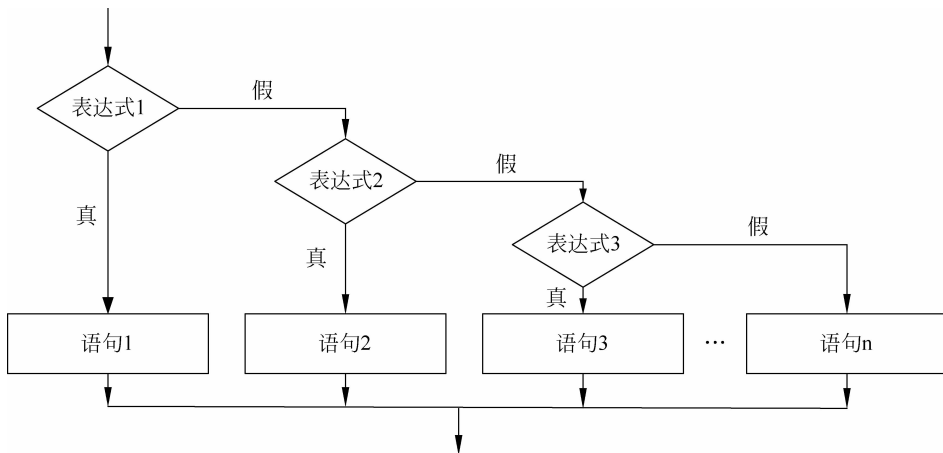


图 3.3 if 语句嵌套形式

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,要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: $\text{if}(x)$ 等价于 $\text{if}(x!=0)$; $\text{if}(!x)$ 等价于 $\text{if}(x==0)$ 。

(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,这时要特别注意 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(图 3.4)

switch 语句的一般形式:

```

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;]
}
  
```

switch 语句在使用过程中需要注意的地方如下:

(1) switch 语句的执行过程为: 计算 switch 关键字后面的表达式的值,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。

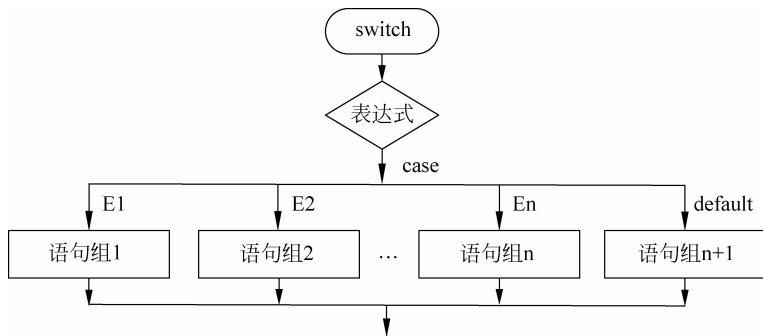


图 3.4 switch 语句

- (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. 学会用选择结构编写简单的程序。

3.3 实验内容

3.3.1 程序设计

1. 编写程序,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 交换;
- (2) 比较 a 和 c,如果 $a < c$,则将 a、c 交换;
- (3) 比较 b 和 c,如果 $b < c$,则将 b、c 交换;
- (4) 按顺序输出 a、b、c。

【流程图】

程序设计 1 的流程图如图 3.5 所示。

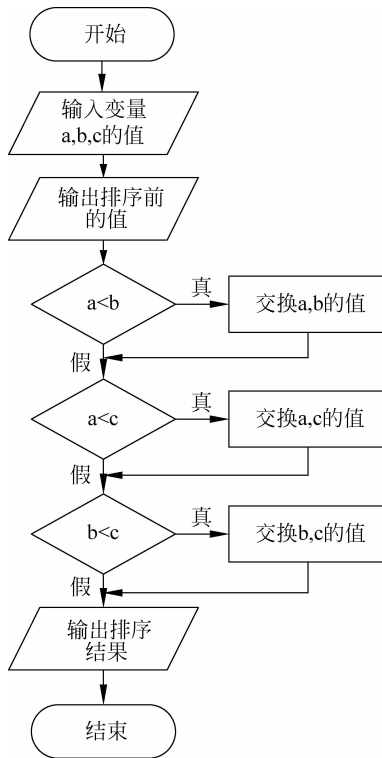


图 3.5 流程图

【参考程序】

```
#include <stdio.h>
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);
}
```

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 计算。

编写程序，输入包裹的重量，输出所需的邮资（输出结果包含两位小数）。

【流程图】

程序设计 2 的流程图如图 3.6 所示。

【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);
}
```

表 3.1

重量/千克	收费标准/元
小于 10	0.80
大于等于 10 但不超过 20	0.75
大于等于 20	0.70

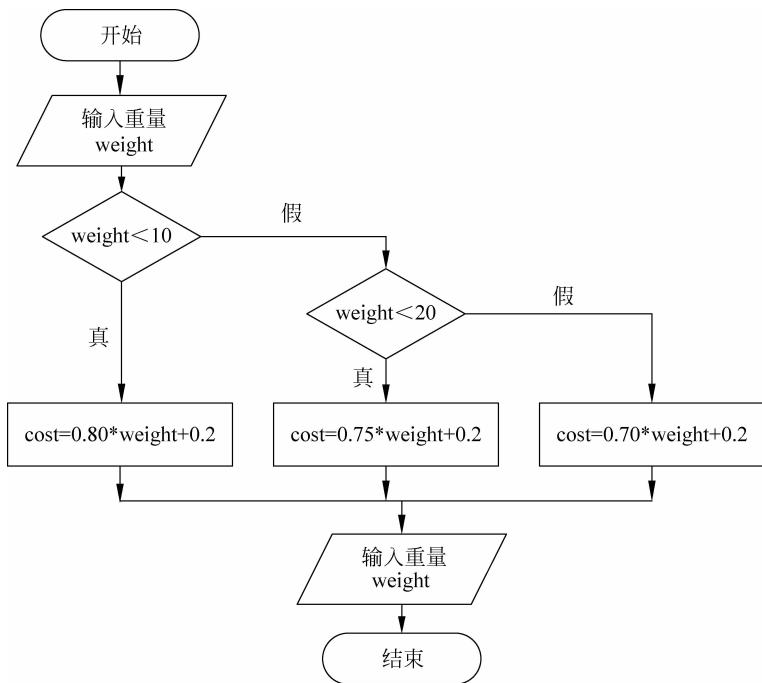


图 3.6 流程图

【说明】

(1) 执行“工具”菜单中的“定制”命令，在如图 3.7 所示的“定制”对话框中选中“调试”选项，显示调试工具条（如图 3.8 所示）。



图 3.7 选择工具栏


(2) 程序调试开始,单击调试工具条中的按钮  (Step Over)。该按钮的功能是单步执行,即单击一次执行一行(如图 3.9 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 3.9 中下方是变量窗口和观察窗口,在观察窗口中可以改变变量的值。



图 3.8 调试工具条

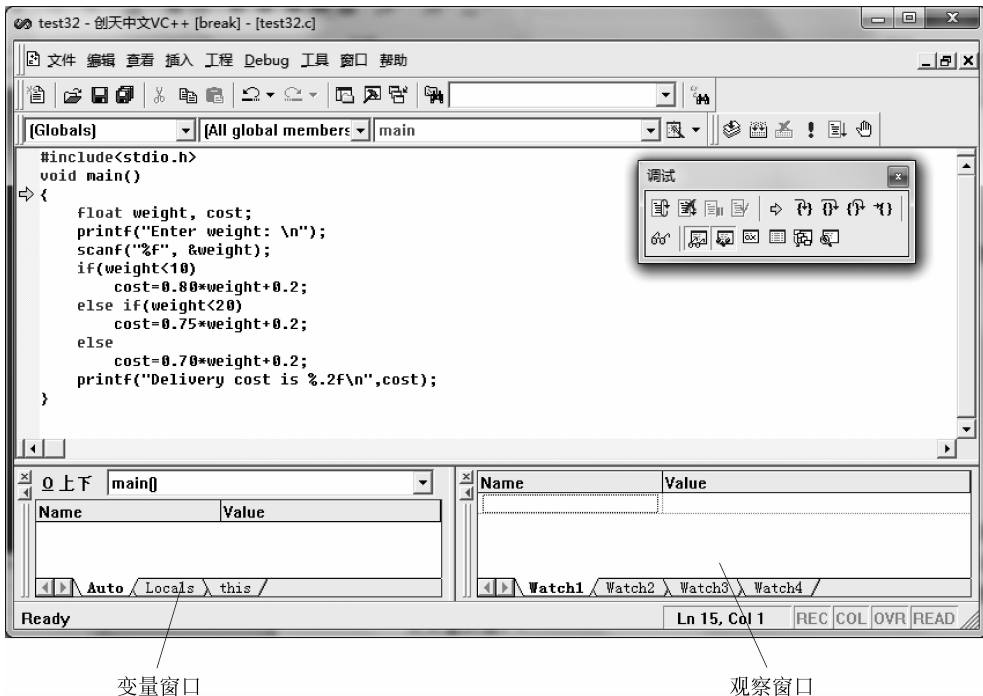




图 3.9 程序调试开始

(3) 单击按钮  (Step Over) 两次, 程序执行到输入语句这一行(图 3.10), 同时运行窗口(图 3.11)显示“Enter weight:”, 继续单击按钮  (Step Over), 在运行窗口中输入“25”(图 3.12), 按 Enter 键后, 箭头指向了“if(weight < 10)”这一行(图 3.13), 在变量窗口中可以看到变量 weight 的值是 25.0000。

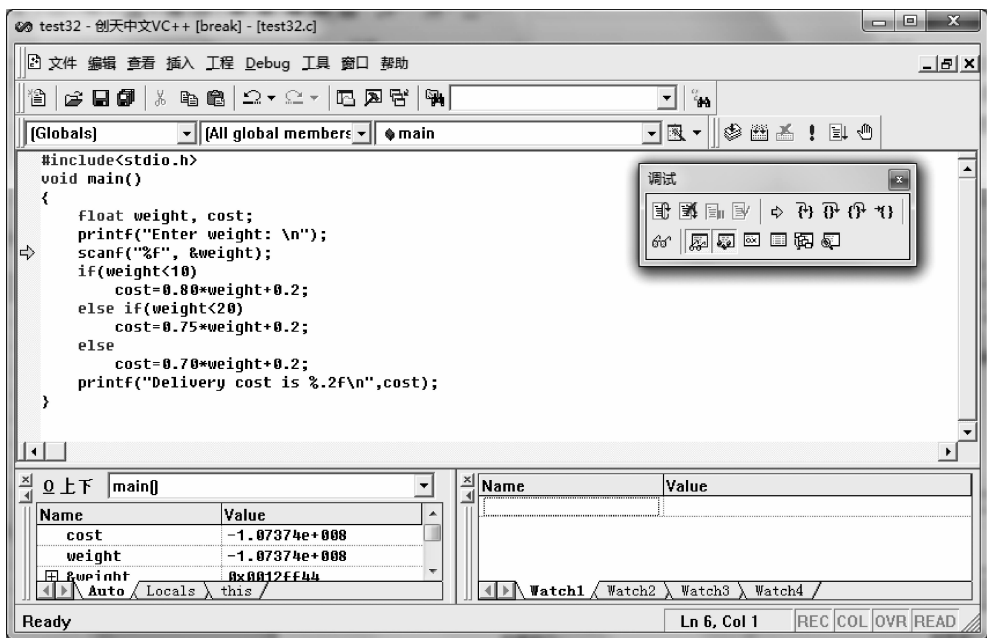


图 3.10 程序单步调试

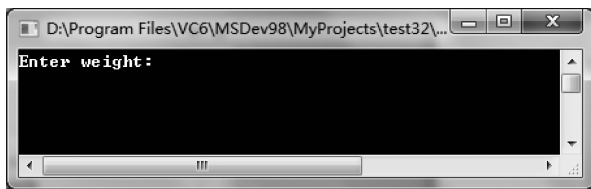


图 3.11 运行窗口

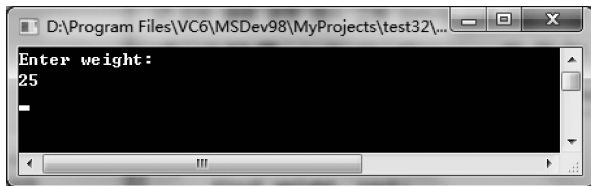




图 3.12 在运行窗口中输入变量 weight 的值

(4) 继续单击按钮  (Step Over) 三次, 箭头指向“printf”这一行(图 3.14), 在变量窗口中可以看到变量 cost 的值为 17.7000。

(5) 继续单击按钮  (Step Over) 一次, 运行窗口显示运行结果(图 3.15)。

(6) 单击按钮  (Stop Debugging), 程序调试结束。

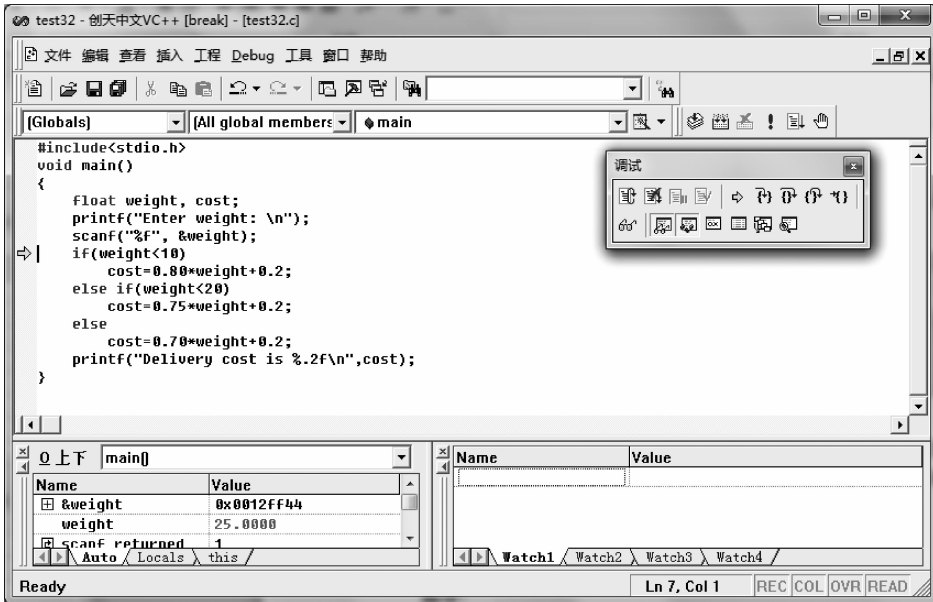


图 3.13 程序单步调试,显示变量 weight 的值

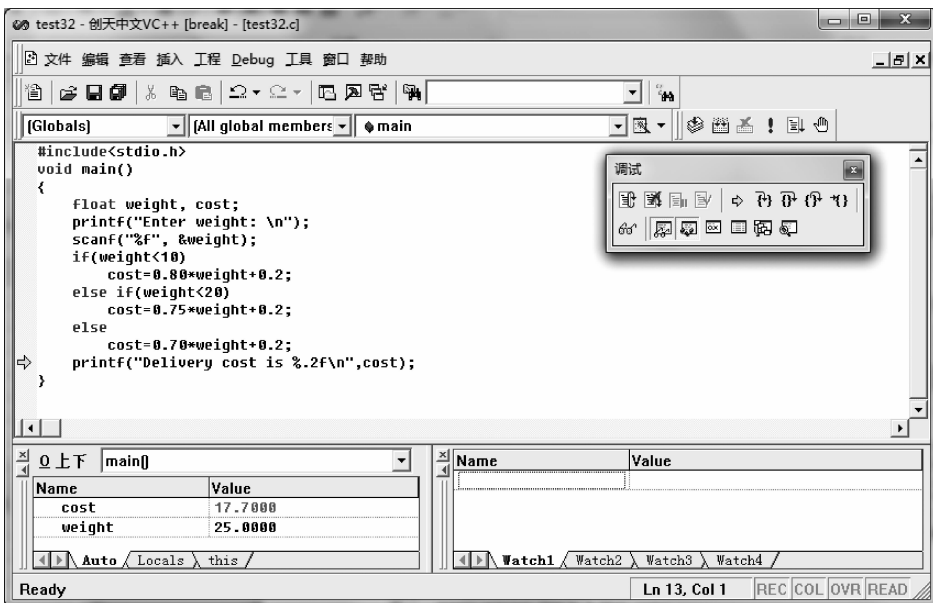


图 3.14 程序单步调试

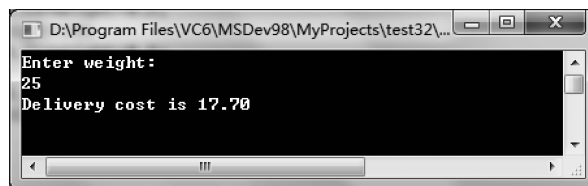


图 3.15 在运行窗口中显示结果

3. 编写程序,其功能是:从键盘输入一个不多于5位的整数,能显示出它是几位数,并按正反两种顺序显示出各位数字。例如,若输入整数1234,输出为:

```
n = 4
1234
4321
```

【指导】

算法分析:首先将该问题分解为三个子问题:一是判断输入的数是几位,二是分离出输入数的各位数字,三是按正序和反序输出各位数字。算法步骤如下:

(1) 用if...else嵌套结构判断输入的整数是几位数,例如,如果 $x > 10000$,则 x 为五位数,即 $n=5$;否则,再继续判断是四位数还是三位数、两位数或一位数。

(2) 根据该数的位数 n ,用switch语句分别求出 x 的每一位数字。例如,当 $n=4$ 时,先求出个位数字,再求出十位数字,直到千位数字,每求出一位数字后,将 x 缩小到原来的 $1/10$,以便求下一位数字。

(3) 按顺序和反序打印出各位数字。

注意:C语言中的int型变量的存储范围。当输入的正整数大于32767时,需要用long int型变量进行存储。

【参考程序】

```
#include <stdio.h>
main()
{
    char c1,c2,c3,c4,c5;           //存放各位数字的变量
    int n;                         //存放整数的位数
    long int x;                    //整数x
    c1 = c2 = c3 = c4 = c5 = ' ';  //给5个变量赋值空格
    scanf("%ld", &x);
    if (x >= 10000)                //确定x是几位数
        n = 5;
    else if (x >= 1000)
        n = 4;
    else if (x >= 100)
        n = 3;
    else if (x >= 10)
        n = 2;
    else
        n = 1;
    switch (n)                     //获取x的各位数字
    {
        case 5: c5 = x % 10 + '0'; x = x/10;
        case 4: c4 = x % 10 + '0'; x = x/10;
        case 3: c3 = x % 10 + '0'; x = x/10;
        case 2: c2 = x % 10 + '0'; x = x/10;
        case 1: c1 = x % 10 + '0';
    }
    printf("n = %d\n", n);
    printf("%c%c%c%c%c\n", c1, c2, c3, c4, c5); //顺序打印
    printf("%c%c%c%c%c\n", c5, c4, c3, c2, c1); //逆序打印
}
```

3.3.2 程序分析

1. 运行下面的程序,分析运行结果。

```
#include <stdio.h>
void 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);
}
```

【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 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≤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≤600 000 元时,高于 400 000 元的部分按 3%提成; 当 600 000 元<I≤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。

提示: 关键是要正确写出计算每一区间奖金的正确公式。例如 20 万元至 40 万元这个区间的奖金由两部分组成:

① 20 万元以上部分应得的奖金为 $(I-200\,000) \times 0.05$ 元;

② 满 20 万元利润应得的奖金数额,为低于(含等于)10 万元的奖金,即 $100\,000 \times 0.1$ 元,加上 10 万元至 20 万元之间的奖金,即为 $100\,000 \times 0.075$ 元。

简便起见,可先把 10 万元、20 万元、40 万元、60 万元、100 万元各关键点的奖金数计算出来。如整 10 万元的奖金数 $Bon1=100\ 000\times 0.1$ 元,整 20 万元的奖金数 $Bon2=Bon1+100\ 000\times 0.075$ 元,整 40 万元的奖金数 $Bon4=Bon2+200\ 000\times 0.05$ 元,整 60 万元的奖金数 $Bon6=Bon4+200\ 000\times 0.03$ 元,其他以此类推。然后再加上各区间超过部分的奖金即可。

使用 switch 语句时,合理的方式是先将输入的利润数额变换为 0 至 10 之间的简单整数,这样便于书写 case 语句。要注意利润数计算公式与 case i ($i=0\sim 10$) 语句的对应关系,以及 break 语句的使用。

以利润数 90 万元为例,对应的奖金数应该按如下表达式计算: $bonus=Bon6+(900\ 000-600\ 000)\times 0.015$ 元。而在书写 switch 语句时,应先将 $I=900\ 000$ 整除 100 000,这样对应到 case 9 语句进行相应的计算。需要考虑当利润数为 60 多万元、70 多万元和 80 多万元时,对应的 case 6, case 7, case 8 语句该如何书写。

2. 输入三角形的三条边 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%,两年 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 9.0 折,H 舱为普通舱 8.5 折,K 舱为普通舱 8.0 折,L 舱为普通舱 7.5 折,M 舱为普通舱 7.0 折,N 舱为普通舱 6.5 折,Q 舱为普通舱 6.0 折,T 舱为普通舱 5.5 折,X 舱为普通舱 5.0 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,结果取两位小数。