第3章 分支结构

C语言中的分支结构有if-else结构和switch结构。if-else是常用的双分支结构，switch是多分支结构。

3.1 if-else结构

3.1.1 if单分支

if单分支语法：

```
if(<表达式>)
{
    <语句块A>
}
```

运行过程：先判断表达式的值是“真”还是“假”，是“真”则运行语句块A，是“假”则什么都不运行。

流程图如图3-1所示。

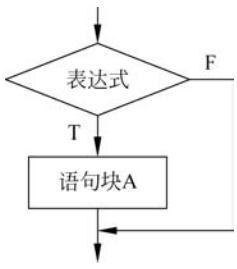

图3-1 单分支流程图

说明：

(1) 表达式一般使用条件表达式或逻辑表达式，表达式必须用一对小括号()括起来。

(2) 语句块A中如果只有一条语句，可以省略一对大括号{}。

(3) 语句块A中如果有多条语句，却没有写大括号{}，则默认只有第一条语句属于分支结构。

3.1.2 if-else双分支

if-else双分支语法：

```
if(<表达式>)
{
    <语句块A>
}
else
{
    <语句块B>
}
```

运行过程：先判断表达式的值是“真”还是“假”，是“真”则运行语句块 A，是“假”则运行语句块 B，两个语句块只能选择一个运行。流程图如图 3-2 所示。

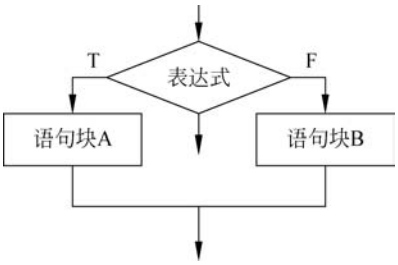

图 3-2　双分支流程图

说明：

（1）不管是语句块 A 还是语句块 B，如果只有一条语句，都可以省略一对大括号{}。

（2）再次说明，如果语句块中有多条语句，必须写大括号{}，否则可能出现逻辑错误。

3.1.3　if-else 嵌套

if-else 嵌套有两种语法，如表 3-1 所示。

表 3-1　if-else 嵌套语法

项目	第一种语法	第二种语法

语法	if(<表达式 1>)
	{
	　if(<表达式 2>)
	　{
	　　<语句块 A>
	　}
	　else
	　{
	　　<语句块 B>
	　}
	}
	else
	{
	　if(<表达式 3>)
	　{
	　　<语句块 C>
	　}
	　else
	　{
	　　<语句块 D>
	　}
	}
		if(<表达式 1>)
		{
		　if(<表达式 2>)
		　{
		　　<语句块 A>
		　}
		　else
		　{
		　　<语句块 B>
		　}
		}
		else if(<表达式 3>)
		{
		　　<语句块 C>
		}
		else
		{
		　　<语句块 D>
		}
		}

说明	else 与 if 都单独一行书写，注意每层语句缩进，以便清晰表示出层次对应关系	将 else 与 if 连写，可以不用多层缩进，但要自己分清 if-else 对应关系

两种语法的流程图是相同的，如图 3-3 所示。

说明：

（1）不管是 if 分支还是 else 分支，都可以再嵌套 if 或 if-else 语句。

（2）允许多层嵌套，但不建议层次太多，否则程序的可读性差，运行效率低。

（3）多层嵌套要注意 if 与 else 的匹配关系，每个 else 总是与在它上面、距它最近且尚未

匹配的 if 配对。

(4) 程序书写应采用缩进方式,将同一层的分支结构对齐,可以增加程序的美观性和可读性。

(5) 避免遗漏大括号造成的程序错误,建议先写好大括号,再编写其中的语句。对于只有一条语句的分支也最好不要省略大括号,避免出现不必要的逻辑错误。

(6) ABCD 四个语句块只能有一个被运行。

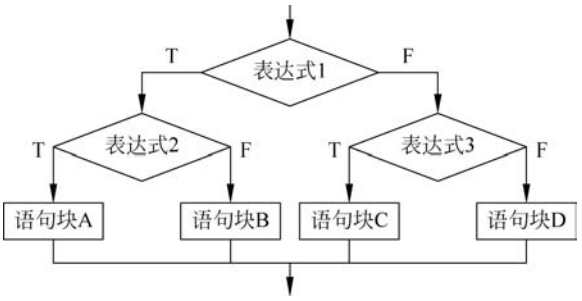

图 3-3 嵌套多分支流程图

3.2 switch 结构

switch 语句是多分支语句结构,但不是所有多分支都能使用 switch 语句。

switch 语法 1:

```
switch(<表达式>)
{
    case <值 1>:语句块 1;
         break;
    case <值 2>:语句块 2;
         break;
    …
    case <值 n>:语句块 n;
         break;
    default:语句块 n + 1;
}
```

语法 1 流程图见图 3-4。

switch 语法 2:

```
switch(<表达式>)
{
    case <值 1>:语句块 1;
    case <值 2>:语句块 2;
    …
    case <值 n>:语句块 n;
    default:语句块 n + 1;
}
```

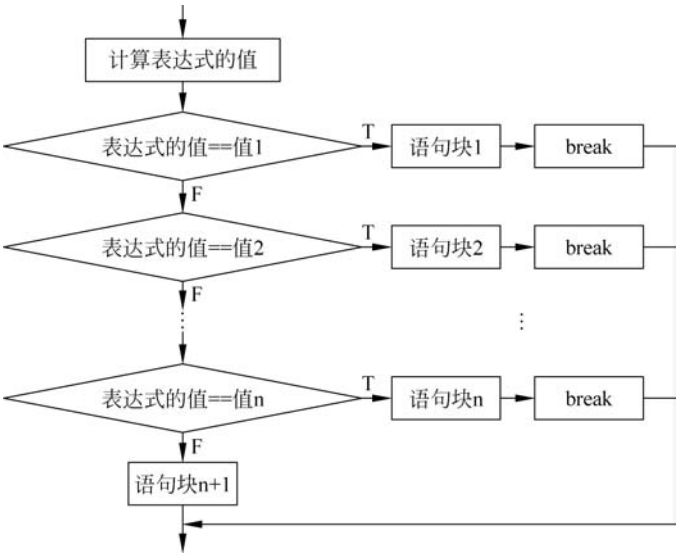

图 3-4　switch 分支流程图 1

语法 2 流程图见图 3-5。

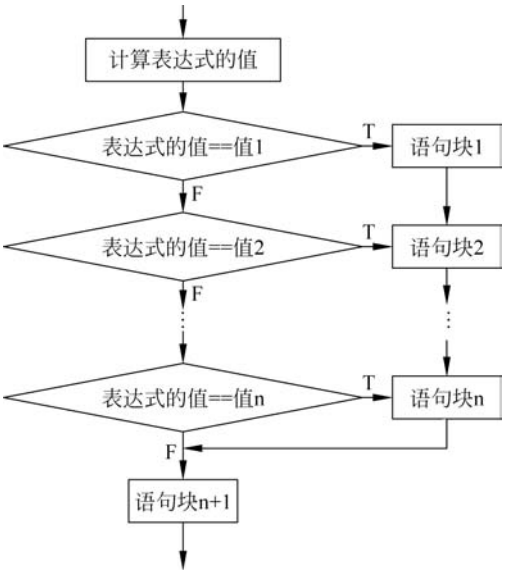

图 3-5　switch 分支流程图 2

说明：

(1) 语法 1 和语法 2 的区别在于 break 语句，有没有 break 语句运行效果明显不同。

(2) case 后的值必须是常量或常量表达式，不可以是变量。

(3) case 后的值必须是整型、字符型或枚举类型，不可以是浮点型或字符串。

(4) case 后的语句块可以是一条语句，也可以是多条语句，但是都不需要用大括号括起来。

(5) 值 1～值 n 必须各不相同，并且要与表达式计算结果的类型一致。

(6) case后的值仅起到标号作用,一旦找到入口标号,就从此标号开始运行,直到遇见break语句,或者运行到switch语句结束,其间不再进行标号判断。

(7) case语句块和default语句块如果都带有break语句,那么它们之间的顺序不影响运行结果,否则运行结果可能会与位置有关。

3.3 程序示例

【例3-1】 输入一个0～100分的成绩,转换为五级制输出,成绩与等级的对应关系如表3-2所示。

表3-2 分数与等级对应表

成绩	成绩<60	60≤成绩<70	70≤成绩<80	80≤成绩<90	成绩≥90
等级	不及格	及格	中等	良好	优秀

参考代码一(使用if-else嵌套语句,程序流程图如图3-6所示)。

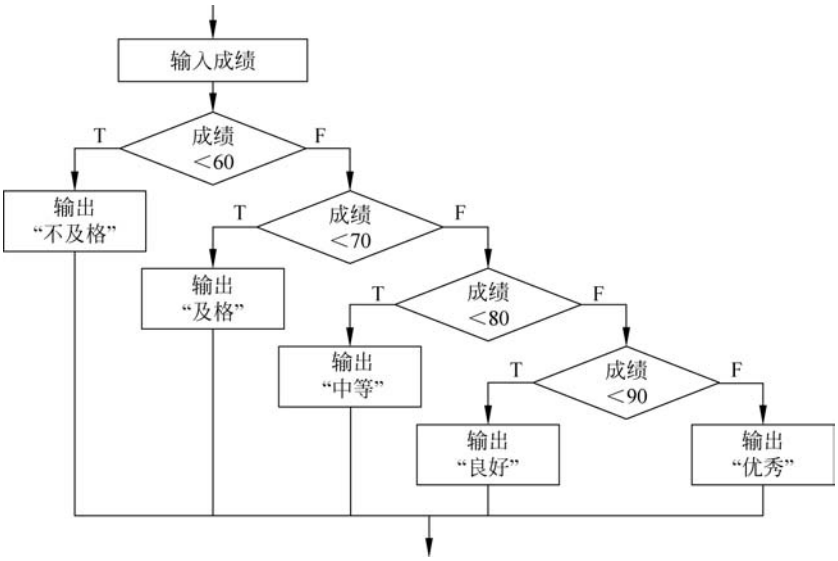

图3-6 成绩等级转换程序流程图

```
#include<stdio.h>
int main( )
{
  int grade;
  printf("请输入分数:");
  scanf("%d",&grade);
  if(grade<60)
        printf("不及格");
  else if(grade<70)
        printf("及格");
  else if(grade<80)
        printf("中等");
```

```
  else if(grade < 90)
        printf("良好");
  else
        printf("优秀");
  getchar( ); getchar( );
}
```

第 1 次运行结果：	第 2 次运行结果：	第 3 次运行结果：
请输入分数：100	请输入分数：78	请输入分数：56
优秀	中等	不及格

程序说明：成绩判断按照由低到高的顺序进行判断。

思考：为什么 else if (grade < 70)不需要写成 else if (grade >= 60 && grade < 70)？

参考代码二（使用 switch 多分支语句）：

```
#include < stdio.h >
int main( )
{
  int grade;
  printf("请输入一个百分制分数:");
  scanf("%d", &grade);
  switch(grade/10)                    //使用 switch 多分支语句,对 grade/10 的值判断
  {
    case 10:                          //100 分与 90 多分都是优秀,灵活利用 break 语句
    case 9: printf("优秀"); break;
    case 8: printf("良好"); break;
    case 7: printf("中等"); break;
    case 6: printf("及格"); break;
    default: printf("不及格");        //不及格涉及的值比较多,放在 default 中
  }
}
```

程序说明：

（1）不是所有多分支都可以使用 switch 结构，case 后必须是具体的离散值，不可以是 grade>90 这样的区间。本题通过 grade/10 表达式，利用整数相除结果为整数的特性，将区间转换为具体值。

（2）合理利用 break 语句的功能，100 分和 90 多分都是优秀，不需要写重复代码。

（3）不及格涉及的值比较多，放到 default 中是个讨巧的方法。

参考代码三（使用 if-else 嵌套语句的第二种写法）：

```
#include < stdio.h >
int main( )
{
  int grade;
  printf("请输入一个百分制分数:");
  scanf("%d",&grade);
  if (grade >= 0 && grade <= 100)          //加入判断成绩是否为 0～100 的语句
  {
    if(grade >= 90)                        //与参考代码一不同,此程序由高分到低分判断
        printf("优秀");
```

```
        else if(grade >= 80)
            printf("良好");
        else if(grade >= 70)
            printf("中等");
        else if(grade >= 60)
            printf("及格");
        else
            printf("不及格");
    }
    else
        printf("成绩应该在 0～100 之间");
}
```

程序说明：

(1) 不管是由高到低，还是由低到高，只要是有序的判断，就可以简写判断条件，如果无序就需要写完整的条件，比如写为：else if (grade >=80 && grade < 90)。

(2) 代码三与代码一同样省略了大括号{}，因为每个分支都是只有一条语句。

(3) 代码三与代码一的流程图不同，自己尝试画流程图。

参考代码四(使用 switch+if-else 语句)：

```
#include <stdio.h>
int main( )
{
    int grade;
    printf("请输入一个百分制分数:");
    scanf("%d", &grade);
    if (grade >= 0 && grade <= 100)          //加入判断成绩是否为 0～100 的语句
    {
        switch(grade/10)                     //使用 switch 多分支语句,对 grade/10 的值判断
        {
        case 10:
        case 9: printf("优秀"); break;
        case 8: printf("良好"); break;
        case 7: printf("中等"); break;
        case 6: printf("及格"); break;
        default: printf("不及格");           //default 分支没有 break 语句,只能放在最后
        }
    }
    else
        printf("成绩应该在 0～100 之间");    //输入分数不符合,直接退出
}
```

程序说明：

(1) switch 结构和 if-else 结构经常配合一起使用，此程序在 if-else 中嵌入 switch 结构，先用 if-else 判断分数是否有效，再对有效值判断等级。

(2) 也可以在 switch 结构的某个 case 分支中嵌入 if-else 结构。

(3) 如果 default 分支也有 break 语句，则 default 分支可以放在任意位置。

3.4 常见错误

错误 1：error C2181：illegal else without matching if(没有匹配 if 的非法 else)。

原因 1：可能 if 分支有多条语句，却没有用大括号{}括起来。

解决办法：将该 else 前面的 if 分支所有语句用大括号{}括起来。建议即使分支中只有一条语句也不省略大括号。

原因 2：可能 if 语句后多了分号，写成了 if(grade＜60)；。

解决办法：去掉分号，正确写法为 if(grade＜60)。

错误 2：error C2061：syntax error(语法错误)，双击错误信息定位到 if 语句。

原因：可能 if 后面的条件没有用小括号()括起来。

解决办法：将 if 语句后面的所有条件都用小括号()括起来。建议先写好一对括号再向括号里填内容。

错误 3：switch 结构关键字书写都正确，编译时却显示一堆错误。

原因 1：可能 switch 后面的条件缺少小括号()，错误示例：switch grade/10。

解决办法：将 switch 语句后面的所有条件都用小括号()括起来。语句修改为 switch (grade/10)正确。

原因 2：可能 switch 后面的条件多了分号，例如写成 switch (grade/10)；。

解决办法：去掉分号，正确写法为 switch (grade/10)。

错误 4：switch 结构中漏掉了 break，造成运行结果错误。

解决办法：在需要退出的地方都加上 break。

错误 5：关系表达式中用错＝和＝＝，一个等号＝是赋值，两个等号＝＝才是比较相等。

解决办法：写完程序要运行检测结果是否正确。

错误 6：复合条件弄错了运算符的优先级。

解决办法：复合条件中加入小括号()，明确优先级。

错误 7：if-else 匹配错误。

解决办法：尽量不省略语句块的大括号，让代码更清晰。每个 else 总是与在它上面、距它最近、且尚未匹配的 if 配对。

实验 4　分支结构练习 1

一、实验目的与要求

1. 掌握 C 语言逻辑值的表示方法(0 表示假，1 表示真)。
2. 学会正确使用关系表达式和逻辑表达式。
3. 掌握各种形式 if 语句语法和使用方法。注意 if 嵌套语句中 if 和 else 的匹配关系。
4. 用 if 语句解决简单的应用问题并上机实现。

二、实验环境

Visual C++ 2010/Visual C++ 6.0。

三、实验内容

1. 编程读入三个整数分别表示箱子的长、宽、高，判断并输出该箱子是正方体还是长方体。

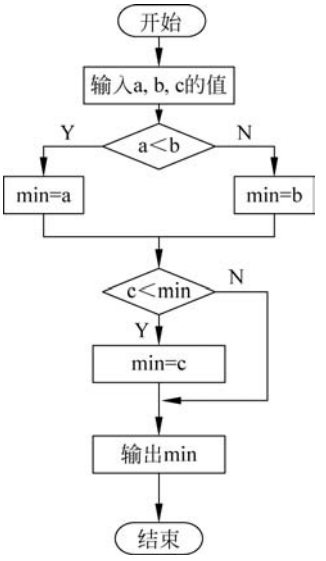

图 3-7　找最小数流程图

提示：判断三个值是否相等需要两两判断，然后进行逻辑运算。

2. 计算表达式$(b+\sqrt{b\times b+2a})/(a-b)$的值，其中，a，b从键盘输入。注意，如果分母等于0，则结果为0。

提示：

(1) 所有变量均定义为双精度浮点数。

(2) 输入数据前显示器上要有提示。

(3) 计算平方根需要用sqrt()函数，该函数在math.h头文件中。

(4) 输出结果保留一位小数。

3. 找最小数：参照如图3-7所示流程图，编写程序输入三个整数，找出最小数并打印输出。

拓展1：请用另外的算法实现第3题程序。

拓展2：编写程序输入三个整数x，y，z，把这三个数由小到大输出。(不是找最小值)

实验5　分支结构练习2

一、实验目的与要求

1. 掌握switch语句语法和使用方法。

2. 掌握switch语句中的break的用法及switch语句的嵌套。

3. 能够用if语句、switch语句解决简单的应用问题并上机实现。

二、实验环境

Visual C++ 2010 / Visual C++ 6.0。

三、实验内容

1. 用switch语句模拟简单的计算器，进行整数的加减乘除四则运算，输入一个表达式，输出表达式的计算结果。例如，输入3*5，输出3*5=15，特殊处理除法，商保留两位小数。

运行效果：

```
请输入表达式:6*9
6*9=54
请输入表达式:17/3
17/3=5.67
```

2. 将第1题改为用if-else实现加减乘除四则运算计算器，比较两个程序的区别，哪一个容易读懂？考虑是否所有程序都可以用两种方法实现。

3. 输入一个时间(整数),时间为 6～10 点,输出“上午好”,时间为 11～13 点,输出“中午好”,时间为 14～18 点,输出“下午好”,其他时间输出“休息时间”。请用 if-else 和 switch 结构分别实现,比较哪一个更好。

运行效果:

```
请输入整数时间: 7
上午好!
请输入整数时间: 12
中午好!
请输入整数时间: 15
下午好!
请输入整数时间: 19
休息时间!
```

拓展:用整数 1～12 表示 1～12 月,由键盘输入一个月份数,输出对应的季节英文名称(12～2 月为冬季,3～5 月为春节,6～8 月为夏季,9～11 月为春季)。要求用 if-else 和 switch 结构分别实现。