第5章

选择结构程序设计

教学目标：

- 掌握6个关系运算符和6个逻辑运算符；
- 学会运算常见的关系表达式和逻辑表达式；
- 熟练掌握逻辑IF语句；
- 熟练掌握块IF结构；
- 灵活使用块IF结构的嵌套；
- 理解块CASE结构。

顺序结构由上而下依次执行每一条语句，只能解决简单的问题。在实际问题中常常要根据不同的条件执行不同的语句，这就需要引入选择结构。

选择结构：根据给定的条件是否成立，选择执行某一部分的操作。故选择结构又叫分支结构。

本章先介绍用于条件准备的关系表达式和逻辑表达式，再介绍用逻辑IF语句、块IF结构和CASE结构来实现选择。

5.1 选择结构中的条件准备

用选择结构，必须先准备条件。准备条件常用关系表达式和逻辑表达式来实现。什么是表达式？表达式就是用运算符将常量、变量和函数等连接起来的式子。运算符是对同类型的数据进行运算操作的符号。表达式的类型由运算符的类型决定。每个表达式按照规定的运算规则产生一个唯一的值。

FORTRAN共有四种表达式，即算术表达式、字符表达式、关系表达式和逻辑表达式。前面章节里我们介绍了算术表达式和字符表达式(字符运算符只有一个“//”字符连接符)，本章介绍关系表达式和逻辑表达式。

5.1.1 关系运算符和关系表达式

1. 关系运算符

FORTRAN95提供了6个关系运算符，如表5.1所示。

使用关系运算符时要注意以下几点。

(1) 两种格式可以单独使用，也可以混合使用。

(2) 使用字母格式时，两边黑点不能省略。

(3) 各关系运算符优先级别相同，从前向后依次运算即可。

表 5.1　FORTRAN95 支持的关系运算符

关系运算符		英 语 含 义	数学意义
字母格式	符号格式		
.lt.	<	less than	小于
.le.	<=	less than or equal to	小于等于
.eq.	==	equal to	等于
.ne.	/=	not equal to	不等于
.gt.	>	greater than	大于
.ge.	>=	greater than or equal to	大于等于

2. 关系表达式

由关系运算符把两个算术量或字符量连接起的式子称为关系表达式。关系运算即“比较运算”。

关系表达式结果是逻辑常量，故关系表达式是最简单的逻辑表达式。一般格式为：

<算术量 1> 关系运算符 <算术量 2>

例如，5>3、A<B、A+B>=C-D、'china'>'canada'、MOD(M,2).EQ.0 等都是合法的关系表达式。

说明：

(1) 关系表达式中，算术量一般是算术型的常量、变量、函数和算术表达式。

(2) 关系运算符两侧还可以是字符型量。如果是字符型量，则称为字符型关系表达式。

(3) 关系表达式的结果是逻辑常量，即.TRUE.或.FALSE.。

逻辑值在内存中用-1(.TRUE.)或 0(.FALSE.)来进行存储，因而从语法而言，逻辑值可以作为关系运算符的算术量，即 6>5>4 是合法的关系表达式。但不主张这样做，因为在 FORTRAN 中，6>5>4 的结果是假，这与数学上的结果不同，易造成错误，这一点要特别注意。实际使用中，一个关系表达式最好只使用一个关系运算符。如果要表示上述 6>5>4 条件，应用逻辑运算符连接，5.12 节将会介绍。

(4) 关系运算符两边的算术量类型不一致时，将自动进行类型转换，转换原则是低级向高级转换。

(5) 谨慎使用等于或不等于关系运算符来判断实型数据之间的关系。由于实型数据在存储时是用近似值表示的，可能存在误差，因此在判定两个实型算术量是否相等时，通常采用差值比较方式。如 A.EQ.B 可以改写为 ABS(A-B)<1E-6 的形式，当 A 与 B 的差值小于某个很小的数(通常取 1E-6)时，可以认为 A 与 B 相等。

(6) 算术运算符的优先级别高于关系运算符。

(7) 算术量是字符串的关系表达式是字符关系表达式。字符串比较大小时，遵循以下原则。

① 单个字符进行比较时，比较的是这两个字符的 ACSII 码值。

例如：

```
'C'>'B'的值为真
'E'>'e'的值为假
```

② 两个字符串比较时，比较的是第一对不同字符的 ACSII 码值。

例如，'CHINA'与'CANADA'比较大小时，先比较两字符串的第一个字符'C'的 ASCII 码，由于相同，接着比较第二个字符'H'与'A'的 ASCII 码，'H'的 ASCII 码是 72，而'A'的 ASCII 码是 65，因此'CHINA'>'CANADA'的结果为真。

③ 若两个字符串中字符个数不相等时，则将较短的字符串后面补空格后再比较。

例如，'the'与'there'比较大小时，前面的三个字符完全相同，无法比较出大小，需要在第一个字符串后加空格，然后与第二个字符的'r'进行比较，由于空格的 ASCII 码值最小，故'the'<'there'的值为真。

【例 5-1】 给出下列关系表达式的值。

关系表达式	运算结果
6 > 4	真(T)
3.0 + SQRT(2.0)> 6	假(F)
.FALSE. = = 0	真(T)
SQRT(3.0)/ = 1.732	真(T)
'THis'<'THIN'	假(F)

5.1.2 逻辑运算符和逻辑表达式

1. 逻辑运算符

逻辑运算符是连接两个逻辑量的运算符，FORTRAN95 提供了 6 种逻辑运算符，如表 5.2 所示。

表 5.2 逻辑运算符及运算规则

逻辑运算符	名称	逻辑运算举例	运 算 规 则
.AND.	逻辑与	A.AND.B	交集，当且仅当 A、B 均为真时，逻辑表达式 A.AND.B 的值才为真，否则为假
.OR.	逻辑或	A.OR.B	并集，A 或 B 之一为真，逻辑表达式 A.OR.B 的值就为真，否则为假
.NOT.	逻辑非	NOT.A	逻辑值 A 取反，A 为真时，逻辑表达式 NOT.A 的值为假，A 为假时，逻辑表达式 NOT.A 的值为真
.EQV.	逻辑等	A.EQV.B	A、B 的逻辑值相同时为真，否则为假
.NEQV.	逻辑不等	A.NEQV.B	A、B 的逻辑值不同时为真，否则为假
.XOR.	逻辑异或	A.XOR.B	A、B 的逻辑值不同时为真，否则为假

注：表 5.2 假设 A、B 均为逻辑变量。

为方便理解，表 5.3 列举了当 A 和 B 的逻辑值为不同组合时各种逻辑运算的结果。

表 5.3 不同组合的逻辑运算值

A	B	.NOT.A	.NOT.B	A.AND.B	A.OR.B.	A.EQV.B	A.NEQV.B	A.XOR.B
真	真	假	假	真	真	真	假	假
真	假	假	真	假	真	假	真	真
假	真	真	假	假	真	假	真	真
假	假	真	真	假	假	真	假	假

逻辑运算符的优先级如下：

.NOT.
.AND.
.OR.
.EQV.　.NEQV.　.XOR.

2. 逻辑表达式

逻辑表达式是用逻辑运算符对逻辑量进行运算的表达式，一般格式为：

<逻辑量 1> 逻辑运算符 <逻辑量 2>

FORTRAN95 提供的逻辑量可以是：逻辑常量、逻辑变量、逻辑表达式和关系表达式（关系表达式的结果为逻辑值）。

例如，TRUE. AND. TRUE.、A. OR. B、2 > 1. AND. 3 < 4、. NOT. (. TRUE.. AND.. FALSE.)等都是合法的逻辑表达式。

运算符不只有一类的表达式为混合表达式。混合表达式的运算次序为：先算术，再关系，最后是逻辑。逻辑的顺序是. .NOT. 、. AND. 、. OR. 的顺序。

【例 5-2】 设 A=4.2，B=5，C=3.5，D=1.0。指出表达式的运算次序和结果。

A >= 0.0 .AND. A+C>B+D .OR. .NOT. .TRUE.

解：该表达式是混合表达式，按以下次序进行计算。

（1）A+C 的值是 7.7

（2）B+D 的值是 6.0

（3）A>=0 的值是. TRUE.

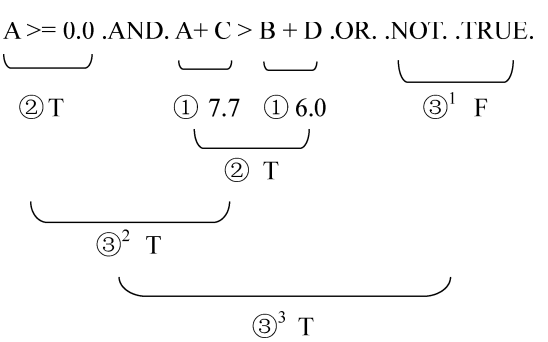
（4）A+C>B+D 即 7.7>6.0 的值是. TRUE.

（5）. NOT. . TRUE. 的值是. FALSE.

（6）A >= 0.0 . AND. A+C>B+D 即. TRUE. . AND. . TRUE. 的值是. TRUE.

（7）A >= 0.0 . AND. A+C>B+D . OR. . NOT. . TRUE. 即. TRUE. . OR. . FALSE. 的值，是. TRUE.

上述过程可以表示为：



合理使用逻辑表达式可以简化判定条件，从而简化语句书写。

5.2 逻辑 IF 语句

逻辑 IF 语句是用来实现最简单的选择结构的语句，一般格式为：

IF(表达式 e)可执行语句 s

逻辑 IF 语句的执行过程是：先计算表达式 e 的值，当表达式 e 的值为真时，执行可执行语句 s，s 执行后，终止该逻辑 IF 语句，继续执行逻辑 IF 语句后面的其他操作；若表达式 e 的值为假，则终止该逻辑 IF 语句，不执行其后可执行语句 s 而直接执行逻辑 IF 语句后面的其他操作。图 5.1 描述了逻辑 IF 语句的执行过程。

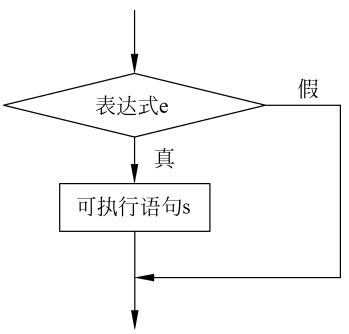


图 5.1 逻辑 IF 语句执行过程

使用逻辑 IF 语句时应注意以下几点。

(1) 逻辑 IF 语句中的可执行语句 s 只能是一条语句。它可以是赋值语句、输入输出语句、STOP 语句等，但不能是 END 语句、其他逻辑 IF 语句、DO 语句、块 IF 语句、ELSE IF 语句、ELSE 语句、END IF 语句和非执行语句。

(2) 表达式 e 的结果必须是一个逻辑值，因此表达式 e 一般是一个关系表达式或逻辑表达式，但也可以是一个整型常量。编译系统将非零整型常量当作.TRUE.，将零当做.FALSE.。

【例 5-3】 输入两个整型数到 A、B 变量，如果 A＞B，输出 A－B 的值，否则结束。

程序编写如下：

```
INTEGER A,B
READ *,A,B
IF(A>B) PRINT*,A-B
END
```

(1) 输入 A＝5，B＝3 时，程序运行结果如图 5.2 所示。

(2) 输入 A＝3，B＝5 时，程序运行结果如图 5.3 所示。

图 5.2 例 5-3 运行结果 1

图 5.3 例 5-3 运行结果 2

由上面的计算结果可以看到，当 A＞B 为假时，则不执行语句“PRINT＊，A－B”。

【例 5-4】 已知三个整数 A、B、C，试编写程序，输出它们的最大值。

流程图见图 5.4。

程序编写如下：

```
INTEGER A,B,C,MAX
PRINT*,"请输入三个整数"
```

```
READ * ,A,B,C
MAX = A
IF(B > MAX) MAX = B
IF(C > MAX) MAX = C
PRINT * , MAX
END
```

程序运行结果如图 5.5 所示。

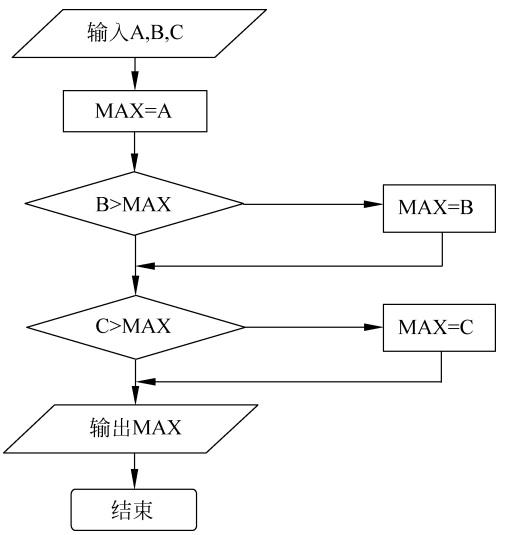


图 5.4　三个数找最大值的流程图

图 5.5　例 5-4 运行结果

5.3　块 IF 结构

逻辑 IF 语句中的可执行语句只有一个，不能描述复杂的操作。若要描述多于一条语句的操作，可用块 IF 结构实现。块 IF 结构分为单分支、双分支和多分支三种情况。

5.3.1　单分支块 IF 结构

一般格式为：

```
IF(表达式 e)THEN
    <THEN 块>
ENDIF
```

块 IF 结构的入口是 IF(表达式 e)THEN 语句，出口是 ENDIF 语句，入口、出口必须配对，结构才完整。

单分支块 IF 结构的执行过程：如果表达式 e 的值为真，则执行 THEN 块，否则什么都不做。THEN 块中可以包含一条或多条可执行语句。

单分支选择结构的流程图见图 5.6。

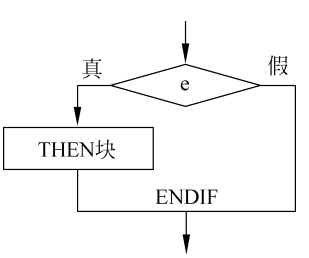


图 5.6　单分支选择结构

【例 5-5】 用单分支块 IF 结构实现例 5-3。

程序编写如下：

```
INTEGER A,B
READ *,A,B
IF(A>B) THEN
  PRINT*,A-B
ENDIF
END
```

程序运行结果如图 5.7 所示。

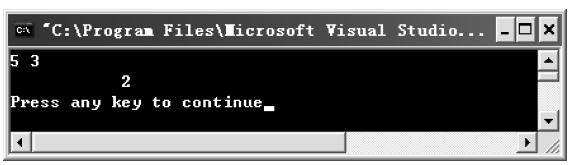


图 5.7 例 5-5 运行结果

【例 5-6】 输入学生的姓名和计算机课成绩，如果成绩大于 60 分，就输出学生的姓名、计算机课成绩和等级合格。

程序编写如下：

```
REAL SCORE
CHARACTER*8,NAME
PRINT*,'请输入学生姓名和计算机课成绩'
READ*, NAME,SCORE
IF(SCORE>60.0)THEN
  PRINT*, NAME,'的计算机课成绩是',SCORE
  PRINT*,'等级成绩：合格'
ENDIF
END
```

程序运行如图 5.8 所示。

如果输入的成绩小于 60 分，则程序运行结果会变为图 5.9。

图 5.8 例 5-6 运行结果 1

图 5.9 例 5-6 运行结果 2

5.3.2 双分支选择块 IF 结构

一般格式如下：

```
IF(表达式 e)THEN
 <THEN 块>
```

```
ELSE
 <ELSE 块>
ENDIF
```

如果表达式 e 的值为真，执行 THEN 块，否则执行 ELSE 块。THEN 块和 ELSE 块都可以包含一条或多条可执行语句。

双分支选择结构的流程图见图 5.10。

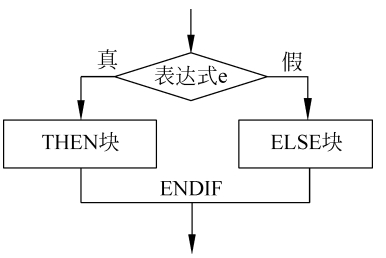


图 5.10　双分支选择结构

【例 5-7】 小学算术减法运算，输入两个整型数到 A、B 变量，判断条件 A>B 是否成立。如果成立则执行 A－B，否则执行 B－A，输出计算结果。

程序编写如下：

```
INTEGER A,B,C
PRINT*,'请输入任意两个整数'
READ*, A,B
IF(A>B)THEN
  C=A-B
ELSE
  C=B-A
ENDIF
PRINT*,"两数之差为:",C
END
```

程序运行结果如图 5.11 所示。

【例 5-8】 输入一个整数，判断它是奇数还是偶数。

程序编写如下：

```
INTEGER NUM
PRINT*,'请输入任意一个整数 '
READ*, NUM
IF(MOD(NUM,2)=  =0)THEN
   PRINT*, NUM,"是一个偶数"
ELSE
   PRINT*, NUM,"是一个奇数"
ENDIF
END
```

程序运行结果如图 5.12 所示。

图 5.11　例 5-7 运行结果

图 5.12　例 5-8 运行结果

5.3.3　多分支块 IF 结构

当需要判断两个或两个以上的条件，即“多重判断”时，需要通过多分支选择结构来

实现。

一般格式如下：

```
IF(表达式 e1)THEN
   <THEN 块 1>
ELSE IF(表达式 e2)THEN
   <THEN 块 2>
ELSE IF(表达式 e3)THEN
   <THEN 块 3>
…
ELSE IF(表达式 en)THEN
   <THEN 块 n>
ELSE
   <ELSE 块>
ENDIF
```

多分支选择块 IF 结构的执行过程是：先计算表达式 e1 的值，当表达式 e1 的值为真时，执行 THEN 块 1，执行完后跳到 ENDIF 语句，结束块 IF 结构；当表达式 e1 的值为假时，计算表达式 e2 的值，当其值为真时，执行 THEN 块 2，执行完后跳到 ENDIF 语句，结束块 IF 结构；否则接着判断表达式 e3 的真假。如此不断重复，直到最后一个条件为真时执行最后一个 THEN 块，否则执行唯一的一个 ELSE 块并结束块 IF 结构。

【例 5-9】 某电视台晚上 9 点的节目安排如下：

星期一、四：卡通片

星期二、五：电视剧

星期三、六：文艺综艺节目

星期日：周日影院

编程实现：当输入星期几时，可查询输出当天晚上的节目。

分析：为简单起见，用整型数 1～7 代表星期一到星期日。

程序编写如下：

```
INTEGER WEEK
PRINT *,"请输入查询数字 1-7,对应查询每日节目:"
READ *,WEEK
IF(WEEK= =1.OR.WEEK= =4) THEN
  PRINT *,"今日节目为:卡通片."
ELSE IF(WEEK= =2.OR.WEEK= =5) THEN
  PRINT *,"今日节目为:电视剧."
ELSE IF(WEEK= =3.OR.WEEK= =6) THEN
  PRINT *,"今日节目为:文艺综艺节目."
ELSE IF(WEEK= =7) THEN
  PRINT *,"今日节目为:周日影院."
ELSE
  PRINT*,"输入查询数字有误!"
ENDIF
END
```

程序运行结果如图 5.13 所示。

图 5.13　例 5-9 运行结果

使用块 IF 结构的说明如下。

(1) 双分支选择块 IF 结构是块 IF 结构的基本结构。一个基本块 IF 结构由 IF 语句、THEN 块、ELSE 语句和 END IF 语句组成。IF 语句、ELSE 语句和 END IF 语句都要单独占一行。

(2) 单分支块 IF 结构缺少 THEN 块或 ELSE 语句、ELSE 块；多分支块 IF 结构则比基本块 IF 结构多 ELSE IF…THEN 语句。不管是哪一种块 IF 结构，IF 语句和 END IF 语句必不可少，如果有 ELSE 语句，则 ELSE 语句和 ELSE 块都只能有一个。

(3) IF 语句行代表块 IF 结构的开始，END IF 语句表示块 IF 结构的结束。块 IF 结构的 THEN 块和 ELSE 块一般由多个可执行语句组成，它们也可以是一个块 IF 结构，这就是 5.4 节要阐述的块 IF 的嵌套。

5.4　块 IF 结构的嵌套

一个块 IF 结构中又完整地包含另一个或多个块 IF 结构，称为块 IF 的嵌套。

使用嵌套的块 IF 结构应注意以下几点。

(1) 在嵌套的块 IF 结构中，内层的块 IF 结构不能和外层的块 IF 结构相互交叉，只能是包含与被包含关系。

为了使程序清晰，一般在书写时应将每一个内嵌的块 IF 结构向右缩进几格，同一层块 IF 结构的 IF 语句、ELSE 语句和 END IF 语句列对齐。

(2) 流程不允许从块 IF 结构外控制转移到块 IF 结构内的任何位置。

一般格式为：

```
IF(..) THEN
  IF(..) THEN
    IF(..) THEN
    ...
    ELSE IF(..) THEN
    ...
    ELSE
    ...
    ENDIF
  ENDIF
ELSE
  IF(..) THEN
  ...
  ELSE
  ...
  ENDIF
ENDIF
```

【例 5-10】 输入学生成绩，按下列条件判断成绩等级：80～100 分为“A”，70～79 分为“B”，60～69 分为“C”，小于 60 分为“D”。

程序编写如下：

```
READ*, GRADE
IF(GRADE>=60) THEN
     IF(GRADE>=70) THEN
        IF(GRADE>=80) THEN
          PRINT*,"A"
        ELSE
          PRINT*,"B"
        ENDIF
     ELSE
      PRINT*,"C"
     ENDIF
ELSE
     PRINT*,"D"
ENDIF
END
```

程序运行结果如图 5.14 所示。

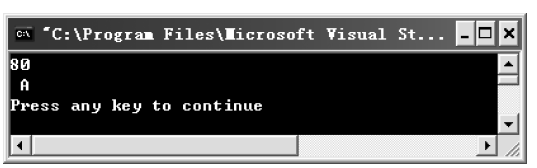


图 5.14 例 5-10 运行结果

用多层块 IF 结构嵌套编写程序比较复杂，可以通过改变判断条件、使用多分支选择结构、改变算法等尽可能减少使用块 IF 结构嵌套的层次。

5.5 块 CASE 结构

写程序时有时会使用“多重判断”，前面已经学习使用块 IF 结构来完成“多重判断”的方法，现在来学习用另一个在语法上更简洁的方法——块 CASE 结构来做这个工作。

块 CASE 结构与多分支选择块 IF 结构非常类似，它可以根据表达式的计算结果，从多个分支中选择一个分支执行。

块 CASE 结构的一般格式为：

```
SELECT CASE(表达式 e)
CASE(数值 1)
   块 1
CASE(数值 2)
   块 2
…
CASE(数值 n)
  块 n
```

```
CASE DEFAULT
  块 n + 1
END SELECT
```

块 CASE 结构的执行过程：首先计算 SELECT CASE 语句中表达式 e 的值 m，接着依次从各数值中寻找 m，如果 m 属于数值 i（$n \geqslant i \geqslant 1$），则执行块 i，并结束块 CASE 结构；若在所有数值内都找不到与 m 相等的常量值，则执行 CASE DEFAULT 下面的块 $n+1$，然后结束块 CASE 结构。

【例 5-11】 用 SELECT CASE 结构实现例 5-9。

程序编写如下：

```
INTEGER WEEK
PRINT *,"请输入查询数字 1 - 7,对应查询每日节目: "
READ *,WEEK
SELECT CASE(WEEK)
CASE (1,4)
  PRINT *,"今日节目为: 卡通片."
CASE (2,5)
  PRINT *,"今日节目为: 电视剧."
CASE (3,6)
  PRINT *,"今日节目为: 文艺综艺节目."
CASE (7)
  PRINT *,"今日节目为: 周日影院."
CASE DEFAULT
  PRINT*,"输入查询数字有误!"
END SELECT
END
```

程序运行结果如图 5.15 所示。

图 5.15　例 5-11 运行结果

使用块 CASE 结构来取代某些多分支块 IF 结构实现多重判断，会让程序看起来更简单，但是使用 CASE 结构有限制，并不是所有的多分支选择结构都能用其来取代。

块 CASE 结构说明如下。

（1）块 CASE 结构从 SELECT CASE 语句开始，到 END SELECT 语句结束。

（2）SELECT CASE 语句中的表达式 e 只能是整型、字符型或逻辑型。

（3）每个 CASE 语句中所使用的数值必须是固定的常量，类型要与表达式 e 的类型一致，不能使用变量。

（4）数值可以是一个常量值，如 5；可以是用逗号“,”间隔的几个常量值，如 1,5,8（共 3 个值）；也可以是采用冒号“:”分隔表示的常量值的区间，如 1:5（表示 1、2、3、4、5 共 5 个值）。

(5) 各 CASE 语句中所使用的数值不能有相同的部分,即块 CASE 结构中表达式 e 的值只能与一个 CASE 语句中的某个常量值相等。

(6) CASE DEFAULT 以及其后的块 $n+1$ 可有可无。如果没有,则在前面所有 CASE 中的数值都与表达式 e 的值不相等的情况下,不执行任何操作。

【例 5-12】 输入两个算术量和算术运算符,输出运算结果。

程序编写如下:

```
INTEGER A,B,C
CHARACTER*2 OPER
PRINT*,'请输入两个整数和一个算术运算符'
READ*,A,B,OPER
SELECT CASE(OPER)
 CASE('+')
    C=A+B
 CASE('-')
    C=A-B
 CASE('*')
    C=A*B
 CASE('/')
    C=A/B
 CASE('**')
    C=A**B
 CASE DEFAULT
  WRITE(*,'("输入运算符不正确")')
 END SELECT
 PRINT*,A,OPER,B,'=',C
 END
```

程序运行结果如图 5.16 所示。

图 5.16 例 5-12 运行结果

5.6 程 序 举 例

【例 5-13】 给定一个学生成绩 S,评判该学生等级,输出结果。假定成绩等级划分如下:

优:95≤S≤100;良:80≤S<95;中:70≤S<80;及格:60≤S<70;不及格:S<60。

分析:为简单起见,假定学生成绩为整数,用整型变量来处理。解决该问题的算法见图 5.17。同一个问题可以用多种选择结构来实现。这里分别采用逻辑 IF 语句、块 IF 结构、块 IF 结构嵌套和块 CASE 结构编写程序,并对它们进行比较。

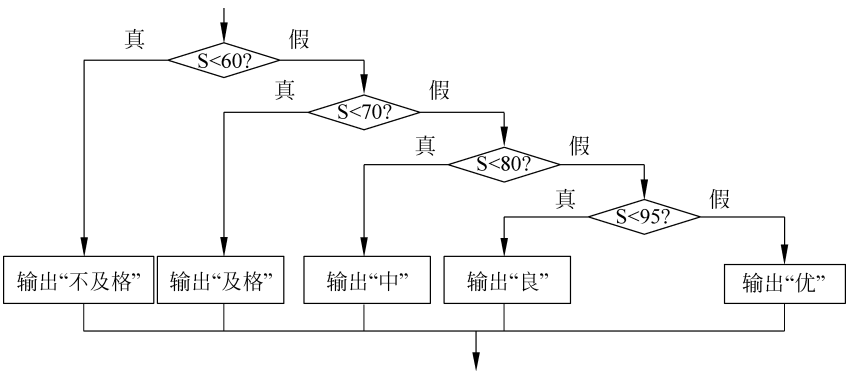


图 5.17　评定学生成绩等级的流程图

程序编写一(采用逻辑 IF 语句实现)：

```
INTEGER S
PRINT *,"输入学生成绩: "
READ *,S
IF (S<60) PRINT *,"该学生成绩为: 不及格."
IF (S>=60 .AND. S<70) PRINT *,"该学生成绩为: 及格."
IF (S>=70 .AND. S<80) PRINT *,"该学生成绩为: 中."
IF (S>=80 .AND. S<95) PRINT *,"该学生成绩为: 良."
IF (S>=95) PRINT *,'该学生成绩为: 优.'
END
```

程序编写二(采用多分支块 IF 结构实现)：

```
INTEGER S
PRINT *,"输入学生成绩: "
READ *,S
IF (S<60) THEN
   PRINT *,"该学生成绩为: 不及格."
 ELSE IF (S<70) THEN
   PRINT *,"该学生成绩为: 及格."
 ELSE IF (S<80) THEN
   PRINT *,"该学生成绩为: 中."
 ELSE IF (S<95) THEN
   PRINT *,"该学生成绩为: 良."
 ELSE
   PRINT *,'该学生成绩为: 优.'
ENDIF
END
```

程序编写三(采用块 IF 结构的嵌套实现)：

```
INTEGER S
PRINT *,"输入学生成绩: "
READ *,S
IF (S<60) THEN
```

```
PRINT *,"该学生成绩为: 不及格."
ELSE
   IF (S<70) THEN
     PRINT *,"该学生成绩为: 及格."
   ELSE
     IF (S<80) THEN
       PRINT *,"该学生成绩为: 中."
     ELSE
        IF (S<95) THEN
          PRINT *,"该学生成绩为: 良."
        ELSE
          PRINT *,'该学生成绩为: 优.'
        END IF
     END IF
   END IF
END IF
END
```

程序编写四(采用块 CASE 结构来实现):

```
INTEGER S
PRINT*,"输入学生成绩: "
READ *,S
SELECT CASE(S)
CASE(0:59)
    PRINT *,"该学生成绩为: 不及格."
CASE(60:69)
    PRINT *,"该学生成绩为: 及格."
CASE(70:79)
    PRINT *,"该学生成绩为: 中."
CASE(80:94)
    PRINT *,"该学生成绩为: 良."
CASE(95:100)
    PRINT *,' 该学生成绩为: 优.'
END SELECT
END
```

以上四段程序中,程序一采用了并列的 5 个逻辑 IF 语句,程序短小简单,可读性高,缺点是由于需要执行每个逻辑表达式,所以运行效率低。

程序三采用块 IF 结构嵌套,运行时只需要计算关系表达式的值,运行效率较高,但程序结构臃肿杂乱。

程序二和程序四运行效率高,且结构层次明晰、简洁,是程序设计的首选。

【例 5-14】 计算下面分段函数的值,编写程序实现。

$$Y=\begin{cases} e^{2\sqrt{|X|}}+\cos X & X<0 \quad (1)\\ 2 & X=0 \quad (2)\\ \dfrac{X}{\sqrt{1+X^2}} & X>0 \quad (3)\end{cases}$$

分析：对于分段函数的计算，首先要判断其自变量的取值范围，根据自变量不同的取值范围来确定执行哪一个计算公式，计算 Y 的值。该类问题求解算法比较简单，使用选择结构实现。

程序编写如下：

```
REAL X,Y
PRINT *,'请输入 X 的值: '
READ *,X
IF(X<0) THEN
  Y=EXP(2*SQRT(ABS(X)))+COS(X)
ELSE IF(X==0) THEN
  Y=2
ELSE
  Y=X/SQRT(1+X**2)
END IF
PRINT *,'Y=',Y
END
```

程序运行结果如图 5.18 所示。

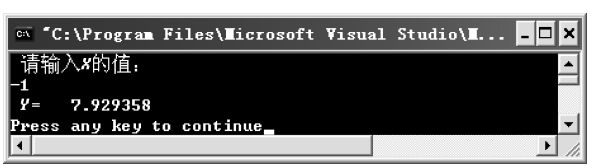


图 5.18　例 5-14 运行结果

【例 5-15】 输入三角形的三条边长 A、B、C，先判断是否构成三角形，若能构成三角形，则计算三角形的三个角 α、β、γ。编写程序实现。

分析：定义三个实型变量，用来存放三角形的三条边，按照三角形的组成规则“任意两边之和大于第三边”建立逻辑表达式：A+B>C. AND. A+C>B. AND. B+C>A。采用反余弦函数计算角度值。

程序编写如下：

```
REAL : : A,B,C,ALFA,BETA,GAMA,X,Y,Z
PRINT*,'请输入三角形三条边的值: '
READ*,A,B,C
IF(A+B>C.AND.B+C>A.AND.C+A>B)THEN
  X=(B**2+C**2-A**2)/(2*B*C)
  Y=(A**2+C**2-B**2)/(2*A*C)
  Z=(A**2+B**2-C**2)/(2*A*B)
  ALFA=ACOSD(X)
  BETA=ACOSD(Y)
  GAMA=ACOSD(Z)
  PRINT*,'角 A=',ALFA
  PRINT*,'角 B=',BETA
  PRINT*,'角 C=',GAMA
ELSE
```

```
  PRINT *,"不构成三角形!"
END IF
END
```

程序运行结果如图 5.19 所示。

图 5.19　例 5-15 运行结果

【例 5-16】 输入任意三个实数 A、B、C,按从小到大的顺序输出。

分析：这是一个简单的排序问题,采用交换算法进行编程。即若 A>B,则 A、B 发生互换,若 A>C,则 A、C 发生互换,这样 A 就是 A、B、C 中的最小者；若 B>C,则 B、C 发生互换,这样 B 就是 B、C 中的最小者。按 A、B、C 顺序打印即可得到从小到大的顺序输出。

程序编写如下：

```
REAL A,B,C,T
PRINT*,'输入任意三个实数给 A,B,C'
READ*,A,B,C
IF(A>B)THEN
  T=A
  A=B
  B=T
ENDIF
IF(A>C)THEN
  T=A
  A=C
  C=T
ENDIF
IF(B>C)THEN
  T=B
  B=C
  C=T
ENDIF
PRINT*,'输入的三个实数按从小到大是:',A,B,C
END
```

程序运行结果如图 5.20 所示。

图 5.20　例 5-16 运行结果

习　题　5

1. 阅读下列程序，给出运行结果。

(1)
```
READ *, N
X = 1.0
IF(N >= 0) X = 2 * X
IF(N >= 5) X = 2 * X + 1.0
IF(N > 15) X = 3 * X - 1.0
PRINT *, X
END
```

如果从键盘输入 15，输出程序运行结果。

(2)
```
READ *, A
IF(A.GE.3.5)THEN
  Y = 3.0
ELSE
    IF(A.GE.4.5)THEN
      Y = 4.5
    ELSE
      Y = 4.0
    ENDIF
ENDIF
PRINT *, Y
END
```

如果从键盘输入 5.0，输出程序运行结果。

(3)
```
LOGICAL P,Q
READ *, X,Y
P = (X.GE.0.0).AND.(Y.GE.0.0)
Q = X + Y > 7.5.AND.(Y >= 0.0)
P = .FALSE.
IF(.NOT.P.AND.Q)THEN
  Z = 0.0
ELSE IF(.NOT.Q) THEN
 Z = 0.0
ELSE IF(P) THEN
 Z = 2.0
ELSE
 Z = 3.0
ENDIF
PRINT *, Z
END
```

如果从键盘输入 3.5，4.5，输出程序运行结果。

(4)
```
CHARACTER A,C
READ *,A
SELECT CASE(A)
CASE('a':'z')
```

```
    C = CHAR(ICHAR(A) - 32)
CASE('A':'Z')
    C = CHAR(ICHAR(A) + 32)
CASE DEFAULT
    C = A
END SELECT
PRINT *,A,C
END
```

如果从键盘输入 E,输出程序运行结果。

2. 填空题

(1) 下面程序判断任意两个整数能否同时被 5 整除,若能,则输出“YES”,否则输出“NO”。请填空。

```
INTEGER M,N
READ( * , * ) M,N
IF( _____ ) THEN
  PRINT *,'YES'
ELSE
  PRINT *,'NO'
_____
END
```

(2) 下面程序判断一个三位整型数是否满足其各位数字之和等于 10。如果满足条件,则输出“YES”,否则不进行任何操作。

```
INTEGER M,I,J,K
READ*,M
I = M/100
J = _____
K = _____
IF(_____) PRINT*,'YES'
END
```

3. 计算职工工资。工人每周工作 40h,超过 40h 的部分应该按加班工资计算(为正常工资的 2 倍)。输入工作时间和单位报酬,计算出该职工应得的工资并输出(假定基本工资为 10 元/h)。

4. 有下列函数:

$$y=\begin{cases}3x-1, & 0\leqslant x<1\\ 2x+5, & 1\leqslant x<2\\ x+7, & 2\leqslant x<3\\ 0, & \text{其他}\end{cases}$$

编写程序,输入 x,输出 y 值。

5. 从键盘输入三个整数,输出其中最大的数。

6. 假定个人所得税有三个等级,且随年龄不同有不同算法。

第 1 类:不满 50 岁

月收入 2000 元以下的税率为 5%，2000～8000 元的税率为 10%，8000 元以上的税率为 15%。

第 2 类：50 岁以上

月收入在 2000 元以下的税率为 3%，2000～8000 元的税率为 7%，8000 元以上的税率为 10%。

编写程序，输入某人的年龄和年收入，计算他(她)一年所应缴纳的税金。