

Nested IF-THEN-ELSE-END IF:

```
IF (logical-expression) THEN
    statements
    IF (logical-expression) THEN
        statements
    ELSE
        statements
    END IF
    statements
ELSE
    statements
    IF (logical-expression) THEN
        statements
    END IF
    statements
END IF
```

Example: Suppose we need a program segment to read a number x and display its sign. More precisely, if x is positive, a + is displayed; if x is negative, a - is displayed; otherwise, a 0 is displayed.

```
IF (x > 0) THEN
    print *, ' + '
ELSE
    IF (x < 0) THEN
        print *, ' - '
    ELSE
        print *, ' 0 '
    END IF
END IF
```

SELECT CASE Statement:

FORTRAN has one more selective execution statement, SELECT CASE, and could be very handy if it is used properly. The SELECT CASE statement is also usually referred to as the CASE statement. The following is its syntactic form:

```
SELECT CASE (selector)  
  CASE (label-list-1)  
    statements-1  
  CASE (label-list-2)  
    statements-2  
  CASE (label-list-3)  
    statements-3  
    .....  
  CASE (label-list-n)  
    statements-n  
  CASE DEFAULT  
    statements-DEFAULT  
END SELECT
```

Where:

1-statements-1, statements-2, statements-3 , ..., statements-n and statements-DEFAULT are sequences of executable statements, including the **SELECT CASE** statement itself.

2- **selector** is an expression whose result is of type INTEGER, CHARACTER or LOGICAL (i.e., no REAL type can be used for the selector).

3- The label lists label-list-1, label-list-2, label-list-3, ..., and label-list-n are called **case labels**,

Each label must be one of the following forms:

Value → 4

value-1 : value-2 → (3 : 7)

value-1 : → (6 :)

: value-2 → (: 9)