3 SELECTION CONSTRUCTS

Selection constructs are used to select between blocks of statements depending on certain *conditions*. Each condition is a logical expression (section 2.4.3) . In FORTRAN, the **IF** statement is used to represent selection constructs. This chapter introduces four types of **IF** constructs: **IF-ELSE**, **IF**, **IF-ELSEIF**, and the **simple IF** constructs.

3.1 IF-ELSE Construct

3.1.1 Definition

The general form of the **IF-ELSE** construct is as follows:

```
IF (condition) THEN
    BLOCK1
ELSE
    BLOCK2
ENDIF
```

where *condition* is a logical expression that evaluates either to .TRUE. or .FALSE.. *BLOCK1* and *BLOCK2* consist of one or more FORTRAN statements. If a block contains more than one statement, each statement must be in a separate line. Statements of *BLOCK1* and *BLOCK2* may be any FORTRAN statements including **IF** statements, assignment statements, input/output statements, repetition statements, transfer (**GOTO**) statements and others. In the above construct, *BLOCK1* will be executed if *condition* has the value .TRUE.. If the value of *condition* is .FALSE., *BLOCK2* will be executed. In either case, only one block is executed. After executing one of the two blocks, control transfers to the first statement after the **ENDIF**.

The keywords **IF** and **THEN** should appear in the same line along with the condition. The condition should be between parentheses. The keyword **ELSE** should appear in a separate line and the construct must end with the keyword **ENDIF** in a separate line. BLOCK1 and BLOCK2 begin, in a new line, after the column in which ***IF***, ***ELSE*** and ***ENDIF*** appear. This is known as *indentation*. Indentation is not a must but it increases program readability.

3.1.2 Examples on the IF-ELSE Construct

The following examples illustrate the **IF-ELSE** construct.

Example 1: *Write a FORTRAN program that reads two integer numbers and prints the maximum.*

Solution:

```
      INTEGER NUM1, NUM2
      READ*, NUM1, NUM2
      PRINT*, 'INPUT: ', NUM1, NUM2
      IF (NUM1 .GT. NUM2) THEN
          PRINT*, 'MAXIMUM IS ', NUM1
      ELSE
          PRINT*, 'MAXIMUM IS ', NUM2
      ENDIF
      END
```

Example 2: *What will be the output of the previous program if the input line is as follows*:

```
    347        -670
```

Solution:

The output will be as follows:

```
INPUT: 347 -670
MAXIMUM IS 347
```

Example 3: *Write a FORTRAN program that reads an integer number and finds out if the number is even or odd. The program should print a proper message.*

Solution:

```
      INTEGER K
      READ*, K
      PRINT*, 'INPUT: ', K
      IF(K / 2 * 2 .EQ. K) THEN
          PRINT*, 'EVEN'
      ELSE
          PRINT*, 'ODD'
      ENDIF
      END
```

Example 4: *What will be the output of the previous program if the input is as follows:*

```
     79
```

Solution: The output will be as follows:

```
INPUT: 79
ODD
```

3.2 IF Construct

3.2.1 Definition

We sometimes require a block of statements to be executed, if a *condition* is .TRUE.. Otherwise, if the condition is .FALSE., no statements must be executed. In this case we use the **IF** construct. The **IF** construct has the following general form:

```
      IF (condition) THEN
         BLOCK
      ENDIF
```

where *condition* is a logical expression that evaluates to either .TRUE. or .FALSE.. *BLOCK* consists of one or more FORTRAN statements. A statement in the *BLOCK* may be any FORTRAN statement including the **IF** statement. *BLOCK* will be executed if the *condition* evaluates to .TRUE. . The control then transfers to the first statement after the **ENDIF**. If the *condition* evaluates to .FALSE., control transfers to the first

statement after **ENDIF**, without executing any statement inside the **IF** construct. The keywords **IF** and **THEN** should appear in the same line along with the condition. The *condition* must be between parentheses. As was the case in the previous **IF** construct, indentation is not a must but it increases readability.

3.2.2 Examples on the IF Construct

The following examples illustrate the **IF** construct.

Example 1: *Write a FORTRAN program that reads a grade. If the grade is not zero, the program must add 2 points to the grade. Then, the new grade should be printed.*

Solution:

```
      REAL GRADE
      READ*, GRADE
      PRINT*, 'ORIGINAL GRADE IS', GRADE
      IF (GRADE .GT. 0) THEN
          GRADE = GRADE + 2.0
          PRINT*, 'SCALED GRADE IS ', GRADE
      ENDIF
      END
```

Example 2: *What will be the output of the previous program if the input line is as follows:*

```
           7.5
```

Solution: The output is as follows:

```
ORIGINAL GRADE IS 7.5000000
SCALED GRADE IS 9.5000000
```

Example 3: *What will be the output of the program of the previous example if the input line is as follows:*

```
           0.0
```

Solution: The output is as follows:

```
ORIGINAL GRADE IS 0.0000000
```

Example 4: *Write a FORTRAN program that reads a student ID and his GPA. If the GPA is greater than or equal to 3.0, the program should print the message 'HONOR'.*

Solution:

```
      REAL GPA
      INTEGER ID
      READ*, ID, GPA
      PRINT*, 'INPUT: ', ID, GPA
      IF (GPA .GE. 3.0) THEN
          PRINT*, 'HONOR'
      ENDIF
      END
```

Example 5: *What will be the output of the previous program if the input line is as follows:*

```
           918962        2.90
```

Solution: The output is as follows: (Note: Since the condition in the **IF** statement is not satisfied, the message HONOR is not printed.)

```
INPUT: 918962 2.9000000
```

3.3 IF-ELSEIF Construct

3.3.1 Definition

Assume you are given a numeric grade. A letter grade is to be printed based on the standard criteria i.e. if the grade is greater than or equal to 90, letter A is to be printed; if the grade is greater than or equal to 80, letter B is to be printed and so on . In such a case, we must use several **IF** statements. Instead FORTRAN provides a construct that can select a single block of statements from several blocks based on different conditions. This construct is the **IF-ELSEIF** construct and it is used when a single block is to be executed from a choice of several blocks. The general form of this construct is as follows:

```
IF (condition-1) THEN
    BLOCK1
ELSEIF (condition-2) THEN
    BLOCK2
ELSEIF (condition-3) THEN
    BLOCK3
..
..
ELSEIF (condition-n) THEN
    BLOCKn
ELSE
    BLOCKn+1
ENDIF
```

where *condition-i* for i = 1, 2, 3, ..., n is a logical expression that evaluates to either .TRUE. or .FALSE.. *BLOCKi* consists of one or more FORTRAN statements. The statements in each BLOCK are FORTRAN statements including any type of **IF** constructs. In the **IF-ELSEIF** construct *BLOCK1* will be executed if *condition-1* evaluates to .TRUE.. The control then transfers to the first statement after the **ENDIF**. If *condition-1* evaluates to .FALSE., condition-2 is examined. If *condition-2* evaluates to .TRUE., *BLOCK2* will be executed and control transfers to the first statement after the **ENDIF**. Otherwise, *condition-3* is examined and if it evaluates to .TRUE., *BLOCK3* will be executed and control transfers to the first statement after the **ENDIF**. The same action is applied to the rest of the **ELSEIF** clauses until a *condition* evaluates to .TRUE.. If all *conditions* evaluate to .FALSE., the **ELSE** part, i.e. *BLOCKn+1*, is executed and control passes to the first statement after the **ENDIF**. The **ELSE** part is optional. If all *conditions* are .FALSE and there is no **ELSE** part, control passes to the first statement after the **ENDIF**, without executing any of the blocks. In summary, the block corresponding to first condition that evaluates to .TRUE. is the only block that is executed. In case, no condition evaluates to .TRUE., the block corresponding to the **ELSE** part, if present, is executed. Indentation is not a must but it increases readability.

3.3.2 Examples on the IF-ELSEIF Construct

The following examples illustrate the **IF-ELSEIF** construct

Example 1: *Write a FORTRAN program that reads a student ID and his GPA out of 4.0. The program should print a message according to the following:*

Condition	Message

GPA ≥ 3.5	EXCELLENT
3.5 > GPA ≥ 3.0	VERY GOOD
3.0 > GPA ≥ 2.5	GOOD
2.5 > GPA ≥ 2.0	FAIR
GPA < 2.0	POOR

Solution:

```
      REAL GPA
      INTEGER ID
      CHARACTER*10 STATE
      READ*, ID, GPA
      PRINT*, 'INPUT: ', ID, GPA
      IF (GPA .GE. 3.5) THEN
          STATE = 'EXCELLENT'
      ELSEIF (GPA .GE. 3.0) THEN
          STATE = 'VERY GOOD'
      ELSEIF (GPA .GE. 2.5) THEN
          STATE = 'GOOD'
      ELSEIF (GPA .GE. 2.0) THEN
          STATE = 'FAIR'
      ELSE
          STATE = 'POOR'
      ENDIF
      PRINT*, ID,' ', STATE
      END
```

Another Solution:

```
      REAL GPA
      INTEGER ID
      CHARACTER*10 STATE
      READ*, ID, GPA
      PRINT*, 'INPUT: ', ID, GPA
      IF (GPA .LT. 2.0) THEN
          STATE = 'POOR'
      ELSEIF (GPA .LT. 2.5) THEN
          STATE = 'FAIR'
      ELSEIF (GPA .LT. 3.0) THEN
          STATE = 'GOOD'
      ELSEIF (GPA .LT. 3.5) THEN
          STATE = 'VERY GOOD'
      ELSE
          STATE = 'EXCELLENT'
      ENDIF
      PRINT*, ID,' ', STATE
      END
```

Example 2: *The following table has two columns, the first column gives the sample input to the previous program and the second column shows the expected output.*

Solution:

Sample Input	Expected Output
927322 2.3	INPUT: 927322 2.3000000 927322 FAIR
922822 3.4	INPUT: 922822 3.4000000 922822 VERY GOOD
848000 1.8	INPUT: 848000 1.8000000 848000 POOR

899999 3.7	INPUT: 899999 3.7000000 899999 EXCELLENT
912877 2.0	INPUT: 912877 2.0000000 912877 FAIR
943245 -2.0	INPUT: 943245 -2.0000000 943245 POOR
942221 7.0	INPUT: 942221 7.0000000 942221 EXCELLENT

Example 3: *Use **IF-ELSE** constructs to write a FORTRAN program that reads a student ID and his GPA out of 4.0. The program should print a message according to the following:*

Condition	Message
GPA ≥ 3.5	EXCELLENT
3.5 > GPA ≥ 3.0	VERY GOOD
3.0 > GPA ≥ 2.5	GOOD
2.5 > GPA ≥ 2.0	FAIR
GPA < 2.0	POOR

Solution:

```
      INTEGER ID
      REAL GPA
      CHARACTER*10 STATE
      READ*, ID, GPA
      PRINT*, 'INPUT: ', ID, GPA
      IF (GPA .GE. 3.5) THEN
          STATE = 'EXCELLENT'
      ELSE
          IF (GPA .GE. 3.0) THEN
              STATE = 'VERY GOOD'
          ELSE
              IF (GPA .GE. 2.5) THEN
                  STATE = 'GOOD'
              ELSE
                  IF (GPA .GE. 2.0) THEN
                      STATE = 'FAIR'
                  ELSE
                      STATE = 'POOR'
                  ENDIF
              ENDIF
          ENDIF
      ENDIF
      PRINT*, ID,' ', STATE
      END
```

Example 4: *Rewrite the above program using **IF** constructs.*

Solution:

```
      INTEGER ID
      REAL GPA
      CHARACTER*10 STATE
      READ*, ID, GPA
      PRINT*, 'INPUT: ', ID, GPA
      IF (GPA .GE. 3.5) THEN
          STATE = 'EXCELLENT'
      ENDIF
      IF (GPA .GE. 3.0 .AND. GPA .LT. 3.5) THEN
          STATE = 'VERY GOOD'
      ENDIF
      IF (GPA .GE. 2.5 .AND. GPA .LT. 3.0) THEN
          STATE = 'GOOD'
      ENDIF
      IF (GPA .GE. 2.0 .AND. GPA .LT. 2.5) THEN
          STATE = 'FAIR'
      ENDIF
      IF (GPA .LT. 2.0) THEN
          STATE = 'POOR'
      ENDIF
      PRINT*, ID,' ', STATE
      END
```

Example 5: *Write a FORTRAN program that reads three integer numbers and finds and prints the maximum. Use* ***IF-ELSEIF*** *construct.*

Solution:

```
      INTEGER X1, X2, X3, MAXIM
      READ*, X1, X2, X3
      IF (X1 .GE. X2 .AND. X1 .GE. X3) THEN
          MAXIM = X1
      ELSEIF (X2 .GE. X3) THEN
          MAXIM = X2
      ELSE
          MAXIM = X3
      ENDIF
      PRINT*, 'THE NUMBERS ARE ', X1, X2, X3
      PRINT*, 'THE MAXIMUM OF THE THREE NUMBERS = ',MAXIM
      END
```

3.4 Simple IF Construct

3.4.1 Definition

Sometimes a single FORTRAN statement must be executed if a *condition* is .TRUE.. In such cases, we may use a simple form of the **IF** construct which is written in a single line. It has the following general form:

```
      IF (condition) STATEMENT
```

where *condition* evaluates to .TRUE. or .FALSE. and STATEMENT is a simple FORTRAN statement such as an assignment statement, a **READ** statement, a **PRINT** statement, a **GOTO** statement, or a **STOP** statement. If *condition* evaluates to .TRUE., *STATEMENT* is executed and the control passes to the next statement. If *condition* is .FALSE., *STATEMENT* is not executed and the control transfers to the next statement.

3.4.2 Examples on the Simple IF Construct

The following examples illustrate the simple **IF** construct.

Example 1: *Use **simple IF** constructs to write a FORTRAN program that reads a student ID and his GPA out of 4.0. The program should print a message according to the following:*

Condition	Message
GPA ≥ 3.5	EXCELLENT
3.5 > GPA ≥ 3.0	VERY GOOD
3.0 > GPA ≥ 2.5	GOOD
2.5 > GPA ≥ 2.0	FAIR
GPA < 2.0	POOR

Solution:

```
      INTEGER ID
      REAL GPA
      CHARACTER*10 STATE
      READ*, ID, GPA
      PRINT*, 'INPUT: ', ID, GPA
      IF (GPA .GE. 3.5) STATE = 'EXCELLENT'
      IF (GPA .GE. 3.0 .AND. GPA .LT. 3.5) STATE = 'VERY GOOD'
      IF (GPA .GE. 2.5 .AND. GPA .LT. 3.0) STATE = 'GOOD'
      IF (GPA .GE. 2.0 .AND. GPA .LT. 2.5) STATE = 'FAIR'
      IF (GPA .LT. 2.0) STATE = 'POOR'
      PRINT*, ID,' ', STATE
      END
```

Example 2: *Write a FORTRAN program that reads three integer numbers and finds and prints the maximum. Use **simple IF** constructs.*

Solution:

```
      INTEGER X1, X2, X3, MAXIM
      READ*, X1, X2, X3
      PRINT*, 'THE NUMBERS ARE ', X1, X2, X3
      MAXIM = X1
      IF (X2 .GT. MAXIM) MAXIM = X2
      IF (X3 .GT. MAXIM) MAXIM = X3
      PRINT*, 'THE MAXIMUM OF THE THREE NUMBERS IS ', MAXIM
      END
```

Another Solution:

```
      INTEGER X1, X2, X3
      READ*, X1, X2, X3
      PRINT*, 'THE NUMBERS ARE ', X1, X2, X3
      IF (X1 .GE. X2 .AND. X1 .GE. X3) PRINT*, 'MAXIMUM IS ', X1
      IF (X2 .GE. X1 .AND. X2 .GE. X3) PRINT*, 'MAXIMUM IS ', X2
      IF (X3 .GE. X1 .AND. X3 .GE. X2) PRINT*, 'MAXIMUM IS ', X3
      END
```

3.5 Exercises

1. What will be printed by the following programs? If an error message is generated, which statement causes the error?

```
1.      INTEGER N, M
        N = 15
        M = 10
        IF (M.GE.N) THEN
            M = M + 1
            IF (N.EQ.M) THEN
                N = N + 5
            ELSEIF (N.GT.0) THEN
                N = N + 10
            ENDIF
            M = M - 1
        ENDIF
        M = M - 1
        PRINT*, M, N
        END
```

```
2.      LOGICAL A, B
        INTEGER EX1, EX2, EX3
        READ*, EX1, EX2, EX3
        A = EX1.LE.EX2.OR.EX2.LE.EX3
        B = EX2+2.GT.EX3*2
        IF (B) THEN
            A = .NOT. A
        ELSE
            B = .NOT. B
        ENDIF
        PRINT*, A, B
        END
```

Assume the input for the program is:

```
40 35 20
```

```
3.      REAL A, B, C
        A = -3
        B = -4.0
        IF (.NOT. A.LT.B) THEN
            C = A - B
        ELSE
            C = A * B
        ENDIF
        PRINT*, C
        END
```

```
4.      REAL A,B
        INTEGER I
        READ*, A, I, B
        IF (A.LT.3.0) THEN
            PRINT*, A+I
            IF (B.LT.2.5) THEN
                PRINT*, B**I
            ENDIF
        ELSE
            PRINT*, A*B*I
        ENDIF
        END
```

Assume the input for the program is:

```
2.5 2 2.5
```

```
5.      INTEGER A, B, C
        READ*, A, B, C
        IF (A.GT.B) THEN
            IF (B.LT.C) THEN
                PRINT*, B
            ELSE
                PRINT*, C
            ENDIF
        ELSE
            PRINT*, A
        ENDIF
        PRINT*, A, B, C
        END
```

Assume the input for the program is:

```
-2 -4 -3
```

```
6.      LOGICAL A,B
        INTEGER K1, K2
        K1 = 10
        K2 = 12
        A = K1.LT.K2
        B = .TRUE.
        IF (A) B = .FALSE.
        PRINT*, A, B
        END
```

```
7.      EEAL A, B
        INTEGER K, L
        READ*, A, B, L, K
        IF (A .GT. B) THEN
            IF (A .LT. L/2) THEN
                PRINT*, 'THURSDAY'
            ELSE
                PRINT*, 'SUNDAY'
            ENDIF
        ELSE
            IF (K/4.GE.B-2) THEN
                PRINT*, 'MONDAY'
            ELSE
                PRINT*, 'TUESDAY'
            ENDIF
        ENDIF
        END
```

Assume the input for the program is:

```
3.0 3.0 4 6
```

```
8.      INTEGER RANKX, RANKY
        REAL X, Y
        READ*, X, Y
        IF (X.GT.Y) THEN
            RANKX = 1
            RANKY = 2
        ELSE
            RANKX = 2
            RANKY = 1
        ENDIF
        PRINT*, RANKX, RANKY
        END
```

Assume the input for the program is:

```
4.0 4.0
```

```
9.      INTEGER SALARY, BONUS, TOTAL
        INTEGER AGE, EXP
        READ*, IDNO, AGE, EXP, SALARY
        IF (AGE.GE.40 .OR. EXP.GT.10) THEN
            BONUS = SALARY/8 + 450.0
        ELSE
            BONUS = SALARY/10 + 350.0
        ENDIF
        TOTAL = SALARY + BONUS
        PRINT*, IDNO, BONUS, TOTAL
        END
```

Assume the input for the program is:

```
834567 38 12 40000
```

2. Write a FORTRAN program that reads the value of a real number (DELTA) . If the value of (DELTA) is negative, then the program prints the message (NUMBER IS OUT OF RANGE) . Otherwise, the program computes the square root of (DELTA) and prints the result.
3. Write a complete FORTRAN program that reads the variables A, B and C, then computes the value of X where:

$$x = \frac{\sqrt{a - b + 2a^2}}{c}$$

   The program should take care of the problem of dividing by zero or getting a negative number under the square root. The program should print the appropriate messages accordingly (i.e. "DIVIDING BY ZERO", or, "NEGATIVE NUMBER UNDER SQUARE ROOT"). If both errors occur, the program should print both messages. If no error occurs, the program should print the value of X.
4. Consider the following structure where A is a real variable :

```
        IF (A.LE.10) THEN
            IF (A.LT.5) THEN
                PRINT*, 'AAA'
            ELSEIF (A.LT.4) THEN
                PRINT*, 'BBB'
            ELSEIF (A.GT.6) THEN
                PRINT*, 'CCC'
            ELSE
                PRINT*, 'DDD'
            ENDIF
        ENDIF
```

The condition that causes AAA to be printed is (A < 5) .

   1. What is the condition that will cause BBB to be printed?
   2. What is the condition that will cause CCC to be printed?
   3. What is the condition that will cause DDD to be printed?

5. Assume that V1 and V2 are **LOGICAL** variables and STATEMENT1, STATEMENT2 and STATEMENT3 are any valid FORTRAN statements. Given the following **IF**-structure:

```
      IF (V1) THEN
          STATEMENT1
      ELSEIF (.NOT. V2) THEN
          STATEMENT2
      ELSE
          STATEMENT3
      ENDIF
```

choose the equivalent structure(s) from the following:

```
I.    IF (.NOT. V1) THEN
          IF (.NOT. V2) THEN
              STATEMENT2
          ELSE
              STATEMENT3
          ENDIF
      ELSE
          STATEMENT1
      ENDIF
```

```
II.   IF (.NOT.V2) THEN
          STATEMENT2
      ELSEIF (V1) THEN
          STATEMENT1
      ELSE
          STATEMENT3
      ENDIF
```

```
III.  IF (V1) THEN
          STATEMENT1
      ELSE
          IF (.NOT. V2) THEN
              STATEMENT2
          ELSE
              STATEMENT3
          ENDIF
      ENDIF
```

6. Consider the following FORTRAN 77 program segment :

```
      IF (A.GT.B .OR. A.EQ.B) PRINT*, A
```

Which one(s) of the following segments is(are) equivalent to the above?

```
I.    IF (A.GE.B) THEN
          PRINT*, A
      ENDIF
```

```
II.   IF (A.GT.B .AND. A.EQ.B) THEN
          PRINT*, A
      ENDIF
```

```
III.  IF (.NOT. (A.LT.B) ) THEN
          PRINT*, A
      ENDIF
```

7. What values of X cause the value of A to be changed in the following statement?

```
       IF (X.LT.3.0 .AND. 7.0.LT.X) A = A + 1
```

8. Write a complete FORTRAN program that reads a real number into a real variable NUM. If NUM is non-zero prints the value of its reciprocal (1/NUM) . Otherwise, prints the message "RECIPROCAL NOT DEFINED".

9. Give the FORTRAN statements that perform the steps indicated below :
    1. If y is not positive, and 3.5>x>1.5 then print the value of y.
    2. If time is greater than 15.0, increment time by 1.0.
    3. If dist is less than 50.0 and time is greater than 10.0, increment time by 2.0. Otherwise, increment time by 2.5.
    4. Interchange the value of a and b (i.e. a gets the value of b and b gets the old value of a, if both a and b are positive.
    5. If grade is greater than or equal to 4.0 then increment a by 1.0. If grade is greater than or equal to 3.0 but less than 4.0 then increment b by 1.0. If grade is greater than or equal to 2.0 but less than 3.0 then increment c by 1.0, otherwise increment d by 1.0.
10. Assume COND1, COND2, COND3, and COND4 are FORTRAN logical expressions. Consider the following program segment.

```
      IF (COND1) THEN
          IF (COND2) THEN
              PRINT*,'RIYADH'
          ELSE
              IF (COND3) THEN
                  PRINT*, 'JEDDAH'
              ELSE
                  PRINT*, 'KHOBAR'
              ENDIF
          ENDIF
      ELSEIF (COND4) THEN
          PRINT*, 'TAIF'
      ELSE
          PRINT*, 'DHAHRAN'
      ENDIF
```

If the output of the above segment is

```
KHOBAR
```

What are the logical values of COND1, COND2, COND3 and COND4?

11. Write a program that reads an integer number N and prints YES if the following expression is satisfied.

    $0 < N < 100$ and $N > 50$

12. Write a FORTRAN program which reads an integer number between 10 and 99 and prints the number reversed. For example, if the number read is 87, then the program output must be 78.
13. Consider the following **IF** statements carefully. Each of Blocks A, B, C, D, E, F, G, H represents a block of FORTRAN statements.

```
I.    IF (CONDITION) THEN
          A
      ELSE
          B
      ENDIF
      C
      END
```

```
II.    IF (CONDITION) D
          E
       END
```

```
III.   IF (CONDITION) THEN
          F
       ELSEIF (CONDITION) THEN
          G
       ELSE
          H
       ENDIF
       END
```

Assuming that X has a value 0.0, which block(s) are executed in program segments (i) , (ii) and (iii) , if CONDITION is the expression listed below?

i) X.GE.0

ii) X.LE.0

iii) X.GT.0

iv) X.LT.0

14. Write a FORTRAN program that reads three integers A, B, and C. The program checks if A, B, and C are in increasing order or in decreasing order and prints an appropriate message. If the integers are not in order, then the program prints UNORDERED. For example, if the input is

```
3 4 5
```

The program prints

```
INCREASING ORDER
```

15. A year between 1900 and 1999 is a LEAP year if it is divisible by 4 and not by 100 or if it is divisible by 400. Write a FORTRAN program which will read a year and determine whether the year is a LEAP or NOT. The program should print one of the following messages accordingly:

```
THE YEAR IS OUT OF RANGE
```

or

```
THE YEAR IS A LEAP YEAR
```

or

```
THE YEAR IS NOT A LEAP YEAR
```

16. Consider the following **IF** statement:

```
       IF (X.GE.Y) THEN
          PRINT*, X
       ELSE
          PRINT*, Y
       ENDIF
```

In each of the following program segments, fill the spaces by relational or logical operators (.EQ., .NE., .LT., LE., .GT., .GE., .AND., .OR., .NOT.) such that each of the program segments below gives the same output as the program segment above.

```
I.     IF (X ------ Y) PRINT*, X
       IF (X ------ Y) PRINT*, Y
```

```
II.    IF (X.GT.Y) THEN
           PRINT*, X
       ELSEIF (X ----- Y) THEN
           PRINT*, X
       ELSE
           PRINT*, Y
       ENDIF
```

```
III.   IF (X ----- Y ----- X.EQ.Y) THEN
           PRINT*, X
       ELSE
           PRINT*, Y
       ENDIF
```

17. Write a program that reads any two positive integer numbers and finds the larger of the two numbers. The program then checks if the larger number is divisible by the smaller one. If it is divisible the program should print the word DIVISIBLE. If the larger number is not divisible by the smaller number, the program checks if both numbers are odd and prints BOTH ODD.

3.6 Solutions to Exercises

Ans 1.

```
 9  15
 F   T
 1.0
  4.5
          -4
          -2  -4  -3
 T  F
 MONDAY
 2   1
 834567   5450   45450
```

Ans 2.

```
      READ*, DELTA
      IF (DELTA .LT. 0.0) THEN
          PRINT*, 'NUMBER IS OUT OF RANGE'
      ELSE
          PRINT*, DELTA ** 0.5
      ENDIF
      END
```

Ans 3.

```
      READ*, A , B , C
      D = A - B + 2 * A ** 3
      IF (C .EQ. 0 .OR. D .LT. 0) THEN
          IF (C .EQ. 0) PRINT*, 'DIVISION BY ZERO'
          IF (D .LT. 0) PRINT*, 'NEGATIVE UNDER SQUARE ROOT'
      ELSE
          X =  D ** 0.5/ C
          PRINT*, X
      ENDIF
      END
```

Ans 4.

1. Never 2. $10 \geq A > 6$ 3. $6 \geq A \geq 5$

Ans 5.

I and III

Ans 6.

I and III

Ans 7.

No values for X,

A can't be changed according to this condition

Ans 8.

```
      REAL NUM
      READ* , NUM
      IF (NUM .NE. 0) THEN
          PRINT*, 1 / NUM
      ELSE
          PRINT*, 'RECIPROCAL NOT DEFINED'
      ENDIF
      END
```

Ans 9.

1.

```
IF( Y .LT. 0 .AND. (X .GT. 1.5 .AND. X .LT. 3.5))PRINT*,Y
```

2.

```
IF( TIME .GT. 15.0 ) TIME = TIME + 1
```

3.

```
      IF( DIST .LT. 50.0 .AND. TIME .GT. 10.0 ) THEN
          TIME = TIME + 2.0
      ELSE
          TIME = TIME + 2.5
      ENDIF
```

4.

```
      IF( A .GT. 0 .AND. B .GT. 0 ) THEN
          T = A
          A = B
          B = T
      ENDIF
```

5.

```
      IF( GRADE .GE. 4.0 ) THEN
          A = A + 1.0
      ELSEIF( GRADE .GE. 3.0 ) THEN
          B = B + 1.0
      ELSEIF( GRADE .GE. 2.0 ) THEN
          C = C + 1.0
      ELSE
          D = D + 1.0
      ENDIF
```

Ans 10.

COND1 : T
COND2 : F
COND3 : F
COND4 : Can be T or F

Ans 11.

```
      READ*, N
      IF (N .GT. 50 .AND. N .LT. 100) THEN
          PRINT*,'YES'
      ENDIF
      END
```

Ans 12.

```
      INTEGER REV
      READ*, K
      IF (K .GT. 10 .AND. K .LE. 99) THEN
          REV = (K - K / 10 * 10) * 10 + K / 10
          PRINT*, REV
      ELSE
          PRINT*, 'NUMBER IS OUT OF RANGE'
      ENDIF
      END
```

Ans 13.

X .GE. 0	i) A , C	ii) D , E	iii) F
X .LE. 0	i) A , C	ii) D , E	iii) F
X .GT. 0	i) B , C	ii) E	iii) H
X .LT. 0	i) B , C	ii) E	iii) H

Ans 14.

```
      READ*, A , B , C
      IF (A .GE. B .AND. B .GE. C) THEN
          PRINT*, 'DECREASING ORDER'
      ELSEIF(A .LE. B .AND. B .LE. C) THEN
          PRINT*, 'INCREASING ORDER'
      ELSE
          PRINT*, 'UNORDERD'
      ENDIF
      END
```

Ans 15.

```
      INTEGER Y
      READ*, Y
      IF(Y .GE. 1900 .AND. Y .LE. 1999) THEN
        IF(Y/4*4.EQ.Y.AND.Y/100*100.NE.Y.OR.Y/400*400.EQ.Y) THEN
          PRINT*, 'THE YEAR IS A LEAP YEAR'
        ELSE
          PRINT*, 'THE YEAR IS NOT A LEAP YEAR'
        ENDIF
      ELSE
          PRINT*, 'THE YEAR IS OUT OF RANGE'
      ENDIF
      END
```

Ans 16.

i) X .GE. Y ii) X .EQ. Y iii) X .GT. Y .OR.X .LT. Y

Ans 17.

```
      READ*, M , N
      IF(M .GE. N) THEN
          MAX = M
          MIN = N
      ELSE
          MAX = N
          MIN = M
      ENDIF
      IF(MAX / MIN * MIN .EQ. MAX) THEN
          PRINT*, 'DIVISABLE'
      ELSE
          IF(MAX/2*2 .NE. MAX .AND. MIN/2*2 .NE. MIN) THEN
              PRINT*,'BOTH ODD'
          ENDIF
      ENDIF
      END
```