Chapter 4
Repetitive Execution

Example: How to calculate the average value of n numbers

$$\frac{a_1 + a_2 + \cdots + a_n}{n}?$$

Especially, if n is large.

Fortran provided repetition structure or loop to do repeated execution of one or more statements.

Counter-controlled DO loops

```
DO control-variable = initial-value, limit, step-size
   statement-sequence
END DO
```

where `initial-value, limit, step-size` are integers.
`step-size` is a nonzero integer. If `step-size` is one, then it can be omitted.

When the loop is executed:

1. The control variable is assigned the initial value.
2. The control variable is compared with the limit to see if it is
   - less than or equal to the limit, for a positive step size
   - greater than or equal to the limit, for a negative step size
3. If so, the statement sequence is executed and the step size is added to the control variable. Otherwise, repetition terminates.

Examples:

```
DO Number = 1, 9
  PRINT *, Number, Number**2
END DO
```

```
READ *, Number
DO I = 1, Number, 2
   Sum = Sum + I
END DO
PRINT *, Sum
```

The do loops can be nested.

```
DO M = 1, Last_M
  DO N = 1, Last_N
     Product = M * N
     PRINT *, M, " X ", N, " = ", Product
  END DO
END DO
```

If the values of Last M and Last N are both 4, then output

```
1 X 1 = 1
1 X 2 = 2
...
2 X 1 = 2
2 X 2 = 4
...
4 X 4 = 16
```

General DO loops

```
DO
  statement-sequence
  IF (logical-expression) EXIT
END DO
```

First do the statement sequence. If the logical expression is false, then repeat the statement sequence.

Example: compute sum of inputs

```
DO
  READ *, Number
  Sum = Sum + Number
  PRINT *, "More numbers to add? (Y or N)?"
  READ *, CONTINUE
  IF (CONTINUE == "N") EXIT
END DO
PRINT *, Sum
```

The DO-CYCLE construct: used to skip some statement conditional in a loop. Statement after CYCLE will be skipped.

```
DO
  PRINT *, "Enter temperature in degree Celsius:"
  READ *, Celsius
  IF (Celsius < 0.0)
    PRINT *, "Temperature must be 0 or above !"
    CYCLE
  END IF
  Fahrenheit = 1.8 * Celsius + 32.0
  PRINT *, Celsius, "degree Celsius = ", &
         Fahrenheit, "degree Fahrenheit"
  PRINT *, "More temperature to convert (Y or N)?"
  READ *, Response
  IF (Response == "N") EXIT
END DO
```

Similar to the select construct, DO construct also can be labelled.

```
A_Loop: DO
  PRINT *, "Enter A:"
  READ *, A
  N_Loop: DO
    PRINT *, "Enter N:"
    READ *, N
    IF (N == 0) THEN
      IF (A == 0)  EXIT A_Loop
      EXIT N_Loop
      PRINT *, A ** N
    END IF
  END DO N_Loop
END DO A_Loop
```

Example

One important statistic used in measuring the reliability of a component in a circuit is the *mean time to failure*, which can be used to predict the circuit's lifetime. Suppose that NASA has awarded an engineering laboratory a contract to evaluate the reliability of a particular component for a future space probe to Mars. As part of this evaluation, an engineer at this laboratory has tested several of these circuits and recorded the time at which each failed; now she would like a program to process this data and determine the mean time to failure.

Analysis and design

Input is a collection of numeric values, but the number of values is unknown.

Output the number of failures and the average of the values.

algorithm:

1. Initialize a counter and running sum to zero.
2. Repeatedly read a new data value, count it, and add it to the running sum.
3. After all the data has been processed, calculate the mean value and display the counter and the mean value.

Step 2 and 3 need to be refined.

Step 2 should use a do loop. Each time read a failure time. We also need a signal to end the loop. For example, to input a -1.0 to stop the loop. So if the value is not -1.0, increase the count and add the value to the running sum. If the value is -1.0, exit the loop.

Step 3 need to consider the situation that the first input is -1.0. In that case, display "no data" message. Otherwise compute the mean value.

Algorithm.

Source code pp. 85.