Repetition Structure

There are two basic types of repetition structures:

- **Loops controlled by a counter:**
  
  the body of the loop is executed once for each value of some control variable in a specified range of values. *DO loop*

- **Loops controlled by a logical expression:**
  
  the decision to continue or to terminate repetition is determined by the truth or falsity of some logical expression. *DO WHILE loop*
In solving problems, we may frequently need to execute some expression /statement repeatedly. Situations where the number of repetitions may be determined in advance is called **Deterministic repetition**.

However, it often happens that the condition to end a repeat structure (or loop) is only satisfied during the execution loop of the loop itself. This type of repeat structure is called **Non-deterministic repetition**.
Iterative or Counting Loop

A loop that executes a block of statements a specified number of times is called Iterative DO loop or counting loop. The counting loop construct has the form

```
DO    index = istart, iend, icr
  Statement 1
  ..........  
  Statement n
END DO
```
Repetition Structure cont’d

- **index** is an integer variable
- **istart** is the initial value index is given
- **iend** is the final value
- **icr** is the increment by which **index** is changed. If it is omitted, unity is assumed

**istart, iend** and **icr** may be positive or negative constants, variables or expressions, but they should always be integers.

```
DO  index = istart, iend, icr
    Statement 1
    ...........
    Statement n
END DO
```
CONTINUE Statement

The CONTINUE Statement is a dummy statement. It is dummy in the sense that it has no effect on the execution of the program. This statement is used primarily as a convenient point for placing a statement label, particularly as the terminal.

```
DO n    index = istart, iend, icr 
    Statement 1
    ...........
    n CONTINUE
```
Example 1: Degrees to radians conversion program

PROGRAM CONVRT
    INTEGER DEGREE
    CONFAC = 3.141593/180.0
    !CONVERSION FACTOR FROM DEGREES TO RADIANS
    DO 10, DEGREE = 0, 360, 10
    RADIAN = DEGREE*CONFAC
    Write(*,*) DEGREE,' degree equals ',Radian, 'radian'
    10 CONTINUE
END
Example 2: Write a program that reads an integer N and computes N!.

```program factorial
implicit none
integer N ! number of iteration
integer :: F = 1
write(*,*) 'GIVE THE NUMBER WHOSE FACTORIAL HAVE TO BE DETERMINED'
read(*,*) N ! Number whose factorial have to be determined
do N = 1, N
   F = F * N
end do
write(*,*) 'FACTORIAL OF', N, 'IS', F
end program factorial```
Nested DO Loop

It is possible for one loop to be completely inside another loop. If one loop is completely inside another one, the two loops are called **nested loops**. The following example shows two nested DO loops used to calculate and write out the product of two integers.

```
PROGRAM NESTED_LOOP
IMPLICIT NONE
INTEGER :: i, j, multiplication
DO  i=1,3
   DO  j=1,3
       multiplication=i*j
       WRITE(*,*) i,'*',j,'=', multiplication
   END DO
END DO
END PROGRAM NESTED_LOOP
```
Non Deterministic Repetition

Sometimes it is not possible to determine the number of iteration count. In such situation, Non Deterministic Repetition is used.

1. **DO: conditional EXIT**

   The general form of DO: conditional EXIT is

   ```
   DO  
     IF (LOGICAL_EXPRESSION) EXIT 
     STATEMENT 1 
     .......
     STATEMENT n 
   END DO 
   ```

   The **EXIT** statement provides a means to exit from an otherwise endless loop. It may in fact go anywhere in the loop. However, it is best for it to go either at the top or at the end.
2. DO WHILE

- A DO construct may be headed with a DO WHILE statement.

```
DO WHILE (Logical_Expression)
    Statement1
    .......
    Statement n
END DO
```

- If the logical expression is true, statements 1 through n will be executed, and then control will return to the DO WHILE statement. This process will be repeated until the logical expression becomes false. When control returns to the DO WHILE statement and the logical expression is false, the program will execute the first statement after the END DO.
3. Alternate for DO WHILE

- **GO TO** statement within IF construct can be used in place of DO WHILE loops:

  n IF (logical-expression) THEN
  statement sequence
  GO TO n
  END IF

- Wherever possible, use an IF construct in preference to GOTO statements.
Example:

```
program dowhile
implicit none
real::x
logical::repeat
repeat=.true.
do while (repeat)
   write(*,*)'enter a real number x,'
   write(*,*)'or a negative number to exit'
   read(*,*)x
   if (x.gt.0.0)then
      write(*,*)'The square root is',sqrt(x)
   else
      repeat=.false.
   end if
end do
end program dowhile
```
Problem 1: Write a program that reads an integer N and computes sum of squares of first N positive integers.

Problem 2: Write a program to find all three digit prime numbers; that is find all prime numbers between 100 and 999
Assignment

Problem 3: Write a program to read a set of numbers, count them, and calculate the mean, variance, and standard deviation of the set of numbers. The mean variance and standard deviation of numbers $x_1, x_2, x_3 \ldots \ldots x_n$ can be calculated using the following formulas:

\[
\text{mean} = \frac{1}{n} \sum_{i=1}^{n} x_i ,
\]

\[
\text{variance} = \frac{1}{n} \sum_{i=1}^{n} x_i^2 - \frac{1}{n^2} \left( \sum_{i=1}^{n} x_i \right)^2
\]

\[
\text{S.D.} = \sqrt{\text{variance}}
\]
Problem 4:

One of the most widely used cubic equation of state is the Soave modification of Redlich-Kwong (SRK) equation:

\[ P = RT(V - b) - \frac{\alpha a}{V(V + b)} \]

Here,

\[ a = 0.42747 \frac{R^2 T_c^2}{P_c} \]
\[ b = 0.08664 \frac{RT_c}{P_c} \]
\[ \alpha = [1 + m (1 - \sqrt{T/T_c})]^2 \]
\[ m = 0.48508 + 1.55171 \omega - 0.15613 \omega^2 \]

Write a program that uses the SRK equation of state to estimate volume of gas for a given temperature and pressure.
BEGIN

ENTER T, D, OMEGA, TC, PC

CALCULATE A, B, M, ALPHA

CALCULATE INITIAL GUESS FOR V USING IDEAL GAS LAW

CALCULATE FV, DFV, VNEW

ABS(VNEW-V) > 1E-5

TRUE

DISPLAY V

END

FALSE