Programming in Fortran Language

The word Fortran is taken from (FORmula TRANSlation) within this language on computer starts from column (1) and ends with column (12), the use of column (1) is only for common statements, while columns (2, 3, 4, 5) are for statement numbers. Column (6) is used for continuation of long statements.

- Any symbol except (0) [Only for long statement]

Col: 1 2 3 4 5 6 7 8

Comment statements number Fortran statements

The Fortran language is composed mainly from letters, numbers, and symbols:

- Letters: A, B, C, ..., Z
- Numbers: 0, 1, 2, ..., 9
- Symbols: / *, :=, =, $, +, -, >, <, \top, \otimes, \oplus, \ominus

Constants:

1. Integer Constant
   \(-6, 2500, 0, -1370, -800\)

2. Real Constant
   \(-0.625, \ldots, 2.5, -0.3766, 7.5\)

- The real constant may appear with an exponent as
  \(314.9 \times 10^{-5}\)
This real constant takes (4 bytes) from the memory and has the value ranged from \((3.4 \times 10^{38})\) to \((1.18 \times 10^{-38})\), and written with no more numbers.

While the double precision real constant takes (8 bytes) from the memory and has the value ranged from \((1.8 \times 10^{-308})\) to \((9.46 \times 10^{-308})\), and written with no more (15) numbers.

\[
\begin{align*}
503.8 \times 10^2 & = 503.8 \\
0.687 \times 10^{-1} & = 0.687 \\
\text{Implicit Double Precision, } \ldots
\end{align*}
\]

3. Character Constant

'Ali' ', Center'

Variables:

The variable name may contain (6) or (8) characters depending on the type of the used computer.

1. Integer Variable

This type of variables takes only the integer values; the integer variable name starts with one of the following letters: \((I, J, K, L, M)\) as:

- \(K200\), \(Moon\), \(Insert\), \(Max\)

2. Real Variables

This type of variables can take only the real values. The real variables name starts with any letter (except the letters \((I, J, K, L, M)\) as:

- \(Small\), \(ZED\), \(Area\)
*Note:

1. You can use an integer variable name to represent a real value using the following instruction:

```
Real the integer variable name
```

Example:
```
Real Num
Num = 10.5
```

2. You can also use a real variable name to represent an integer value.

```
Integer the variable name
```

Example:
```
Integer Area
Area = 10
```

3. Character Variable

This type of variables takes the characters instead of numbers and can be written in the following forms:

```
Character A1, A2, A3
Character *:1 A1, A2, A3
Character A1 *:1, A2 *:1, A3 *:1
```

Example:
```
Character *:6 Var *:3, IN, Age *:20
```

Here in the variable, Age takes 20 bytes.
```
the variable IN    = 6 bytes
```
```
Var    = 3
```
Since there are two types of numbers (real and integer), therefore, there are two types of numerical calculations:

a) Integer Calculations
   \[
   \begin{align*}
   5 - 3 & = 2 \\
   5 \times 3 & = 15 \\
   5 + 3 & = 8 \\
   -7/2 & = -3 \\
   4/5 & = 0 \\
   5/4 & = 1 \\
   \end{align*}
   \]
   \[
   \text{\(I/I\): (Integer/Integer = Integer)}
   \]

b) Real Calculations
   \[
   \begin{align*}
   5.0 - 3.0 & = 2.0 \\
   5.0 \times 3.0 & = 15.0 \\
   5.0 + 3.0 & = 8.0 \\
   -7.0/2.0 & = -3.5 \\
   4.0/5.0 & = 0.8 \\
   5.0/4.0 & = 1.25 \\
   \end{align*}
   \]

*In the use of integer calculation, you must be using parentheses as given below.*
Example:

Find the value of $A$ for odd and even values of $N$

\[ A = \frac{N \times 2}{2} \]

1. When $N$ is an even number, say $N = 4$
   \[ A = \frac{(N \times 2)}{2} \Rightarrow \frac{(4 \times 2)}{2} = 4 \quad \text{o.k.} \]
   \[ A = \frac{(N/2) \times 2}{2} \Rightarrow \frac{(4/2) \times 2}{2} = 1 \quad \text{o.k.} \]

2. When $N$ is an odd number, say $N = 5$
   \[ A = \frac{(N \times 2)}{2} \Rightarrow \frac{(5 \times 2)}{2} = 5 \quad \text{o.k.} \]
   \[ A = \frac{(N/2) \times 2}{2} \Rightarrow \frac{(5/2) \times 2}{2} = 2.5 \quad \text{Not o.k.} \]

Arithmetic Expressions

The arithmetic expression uses the mathematical operations and variables and constants. The sequence of these operations is as follows:

1. (ok *) First
2. (* , /) Second
3. (+, -) Third

Example: Write the following expression in Fortran language

\[ y = \left\lfloor \frac{ab}{c+d} - \frac{g}{(5gh+x)} \right\rfloor^+ \]

Solution:

\[ y = \left( \frac{(a\times b)}{(c+d)} \right) - \left( \frac{g}{(5.0 \times (h+x))} \right)^{** \left(1.0/r \right)} \]

\[ \left(\frac{11}{2} \right) + \frac{4}{3} \]

\[ \left(\frac{1}{2} \right) + \frac{4}{3} \]

\[ 2 \frac{3}{11} = \frac{43}{3} + \left( \frac{5.5 \times 5}{5} \right) = \frac{11}{2} \]

\[ 2.3 = \frac{43}{3} + (5.5 \times 5) = \frac{11}{2} \]

\[ (1.0/r) \]

\[ (5.0 \times (h+x)) \]

\[ \left\lfloor \frac{ab}{c+d} \right\rfloor - \left\lfloor \frac{g}{(5gh+x)} \right\rfloor^+ \]
OR you can write the expression as follows:

\[ T_1 = \frac{a \times b}{(c+d)} \]
\[ T_2 = \frac{g^2}{(5.0 \times (h+x))} \]
\[ y = (T_1 - T_2) \times (1.0/R) \]

H.W. (1): Write the following expressions in Fortran language

1. \( a + \frac{b}{c} + d \)
2. \( \frac{a + b}{c + d} \)
3. \( a^3 - b^3 \)
4. \( \frac{ab}{c} - d^2 \)
5. \( \frac{a + b}{c} - d^2 \)
6. \( \frac{a}{bc} - d^2 \)
7. \( 1 + \frac{a}{b + \frac{1}{c}} \)
1. Unconditional Goto Statement:
   \[ \text{Goto } n \]  
   \( n \) is a statement number

2. Conditional Goto Statement:
   \[ \text{Goto } (n_1, n_2, n_3, \ldots, n_m), J \]  
   \( J \) as:
   \[ \text{Goto } (10, 15, 20, 25), J \]

   when \( J = 1 \) \[ \Rightarrow \] Goto Statement No. 10
   when \( J = 2 \) \[ \Rightarrow \] Goto Statement No. 15
   when \( J = 3 \) \[ \Rightarrow \] Goto Statement No. 20
   when \( J = 4 \) \[ \Rightarrow \] Goto Statement No. 25

3. Arithmetic IF Statement
   \[ \text{IF } (x) \text{ and } n_1, n_2, n_3, \ldots \]  
   \( x \) is an arithmetic expression
   when \( (x) \) is a (negative) value then Statement No. 10
   when \( (x) = 0 \) \[ \Rightarrow \] Goto Statement No. 20
   when \( (x) > 0 \) \[ \Rightarrow \] Goto Statement No. 30

\[ \text{READ } x, x \]
\[ \text{IF } (x) \geq 4 \]
\[ 2 \]
\[ \text{GO TO } 20 \]
\[ 3 \]
\[ \text{GO TO } 20 \]
\[ 4 \]
\[ \text{PRINT } x \]
\[ 5 \]
\[ \text{STOP} \]
\[ 6 \]
\[ \text{END} \]
4. Logical IF Statement

IF (Condition) Statement 1
Statement 2

Relational Operators
- GT.
- GE.
- LT.
- LE.
- EQ.
- NE.

Logical Operators
- AND.
- OR.
- NOT.

5. IF .... Then Statement

IF (Condition) Then
Statement 1
Statement 2
Statement n
End IF.

17 \[ y = x + 1.0 \]

17 \[ y = x + 1.0 \]
Example: Write a program to compute the value of \( y \) from the following:

\[
y = \begin{cases} 
  x & \text{when } x > 0 \\
  -x & \text{when } x \leq 0 
\end{cases}
\]

Solution:

Read \( *, \ x \)
IF \(( \ x \ . \ GT \ . \ 0.0 \ ) \) Then
\( y = x \)
Print \( *, y \)
END IF
\( y = -x \)
Print \( *, y \)
Stop
End

5. IF \( \) \( \) Then \( \) \( \) Else \( \) \( \) Statement

IF (Condition) Then

[ ] Yes
Else

[ ] No
End IF
Example: Write a program to compute and print out the value of \( w, y, z \) from:

\[
\begin{align*}
  w &= \sqrt{ax}, & y &= x^2 - A/3, & z &= A\times x^2 + 5 \quad \text{when } A \leq 0 \\
  w &= 7, & y &= 3, & z &= 2 \quad \text{when } A > 0
\end{align*}
\]

Solution:

```
Read x, A, x
IF (A . LE. 0.0) Then
  w = SQRT(A*x)
  y = x**2 - A/3.0
  z = A*x**2 + 5.0
  Print *, w, y, z
Else
  w = 7.0
  y = 3.0
  z = 2.0
  Print *, w, y, z
End IF
Stop
End
```

7. Nested IF Statement

```
IF (Condition1) Then
  Yes 1
ELSE IF (Condition2) Then
```

ELSE IF (Condition N) Then
    ELSE
        No
    End IF

Example: Write a program to read in the values of Velocity (V), Dynamic Viscosity (Nu) and the Pipe Diameter (D). Then compute (RE) from the following, also define the type of flow.

\[ RE = \frac{VD}{Nu} \]

when \( RE \leq 1200 \), The type of flow is Laminar
when \( 1200 < RE < 4000 \), The type of flow is Transition
when \( RE \geq 4000 \), The type of flow is Turbulent

Solution:

Character * 10 Type
Real V, Nu, D, Re
Read *, V, Nu, D
Re = V*D/Nu
IF (Re .LE. 1200.0) Then
    Type = 'Laminar'
ELSE IF (Re .LT. 4000.0 .OR. GT. 1200.0) Then
    Type = 'Transition'
ELSE IF (Re .GE. 4000.0) Then
    Type = 'Turbulent'
ELSE
Type = 'Undefined'
End IF
Print *, Re, Type
Stop
End

H. W. (2):

Write a program to read in the values of a, b, c, then compute the root values (x) of the equation:

\[ ax^2 + bx + c = 0 \]

\[ x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]
Direct Output Statement

Print *, variable name

the star indicates the direction printing without introducing the sig and location of the printed value

Example:

\[
\begin{align*}
X &= 10.5 \\
y &= 2.3 \\
z &= x + y \\
\text{Print *, 'z = ', z} \\
\text{Stop} \\
\text{End}
\end{align*}
\]

RUN \quad Z = 12.8

\[
\begin{align*}
X &= 10.5 \\
y &= 2.3 \\
l &= x + y \\
\text{Print *, 'l = ', l} \\
\text{Stop} \\
\text{End}
\end{align*}
\]

RUN \quad L = 13

Direct Input Statement

\[\text{Read *, variable name}\]
**Comment Statement**
This statement is not an executable statement and used only for explaining the program steps.

```fortran
C This is a Fortran Program
C To compute the square root of a number x
C
Read *, x
y = SQRT(x)
C *********
C Print the result
Print *, y
Stop
End
```

**Stop Statement**
This statement is an executable statement, it will stop the program whenever the program reaches it, and it may appear more than one time in the program.
<table>
<thead>
<tr>
<th>Library Functions in Fortran</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( \sqrt{x} ) \quad \text{SQRT}(x)</td>
</tr>
<tr>
<td>2. (</td>
</tr>
<tr>
<td>3. ( e^x ) \quad \text{EXP}(x)</td>
</tr>
<tr>
<td>4. ( \sin x ) \quad \text{Sin}(x)</td>
</tr>
<tr>
<td>5. ( \cos x ) \quad \text{Cos}(x)</td>
</tr>
<tr>
<td>6. ( \sin^{-1} x ) \quad \text{ASin}(x)</td>
</tr>
<tr>
<td>7. ( \cos^{-1} x ) \quad \text{ACos}(x)</td>
</tr>
<tr>
<td>8. ( \log x ) \quad ALog_10(x)</td>
</tr>
<tr>
<td>9. ( \ln x ) \quad ALog(x)</td>
</tr>
</tbody>
</table>

Functions (FLOAT) and (IFIX)

* The function (FLOAT) transforms a specified integer value to the corresponding real value.

\[
\text{FLOAT}(4) = 4.0 \\
\text{FLOAT}(-25) = -25.0
\]

* The function (IFIX) transforms a specified real value to an integer value.

\[
\text{IFIX}(2.9) = 2 \\
\text{IFIX}(-3.7) = -3
\]
Do Statement

Do n I = e₁, e₂, e₃

n: number of last statement in the loop
I: Loop indicator
e₁: indicates initial value
e₂: indicates final value
e₃: step size

Example: Write a program to compute the summation value of \((S=0)\)num

Solution:

```
Sum=0.0
Do 10 I = 1, 50
  Read A, X
  Sum = Sum + X
  Continue
10 Print A, 'sum of the numbers=', Sum
Stop
End
```

OR

```
Sum=0.0
Do 10 I = 1, 50
  Read A, X
10 Sum = Sum + X
Print A, 'sum of the numbers=', Sum
Stop
End
```
1. The last statement in the loop cannot be a control statement like (; GoTo, Do, Stop, etc).

2. When IF statement appears inside (Do loop) the (IF block) must be completed inside the loop as:

   ```
   Do 10 K = 1, 8
   IF (SQR T(X).GT. 12.0) Then
   
   Elses
   
   End If
   *10 Continue
   ```

3. When Do statement appears inside (IF block), the (Do Loop) must be completed inside the block as:

   ```
   IF (A.LT. B) Then
   Do 100 I = 1, 10
   100 Continue
   Else
   Do 200 I = 1, 20
   
   200 Continue
   End If
   ```
4. It is allowed to go out of the loop before it completed as:

```plaintext
Do 5 I = 1, 10
   Read X, Y
   X = .Float(I)
   IF ( SQRT(Y) GT SQRT(X) ) Go to 20
   Print X, Y
   Continue
5    Stop
20   End
```

5. You can never inter inside the loop from a statement outside the loop.

Nested Do Loops

```plaintext
10   Do I = 1, 2
20   Do J = 1, 3
25   Print X, I, J
20     Continue
10     Continue
25   Stop
10   End
```

* You can also end the nested loops in one statement as:

```plaintext
Do 10 I = 1, 5
Do 10 J = 1, 7
Do 10 K = 1, 9, 2
10   Continue
```
One-Dimensional Arrays

Array Name (Subscript)
as:

<table>
<thead>
<tr>
<th>Array Name X</th>
<th>Value of X</th>
</tr>
</thead>
<tbody>
<tr>
<td>X(1)</td>
<td>20.3</td>
</tr>
<tr>
<td>X(2)</td>
<td>15.2</td>
</tr>
<tr>
<td>X(3)</td>
<td>-1.6</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>X(10)</td>
<td>-0.5</td>
</tr>
</tbody>
</table>

* The array name may be any variable (integer, real).
* The subscript must be any integer number.
* The array value may be any constant.

Dimension Statement

Dimension array name (largest possible value for the subscript)

Example: Write a program to read in the element values of array X(10) and Y(10), then evaluate the value of Z:

\[ Z(I) := X(I) + Y(I) \]

Solution:

```
Dimension X(10), Y(10), Z(10)
Do 10 I = 1, 10
Read X(I), Y(I)
10 Continue
Do 5 J = 1, 10
```
Z(J) = X(J) + Y(J)
Continue
Do 20 K = 1, 10
Print *1, Z(K)
Continue
Stop
End

OR

Dimension X(10), Y(10), Z(10)
Do 70 I = 1, 10
Read *1, X(I), Y(I)
Z(I) = X(I) + Y(I)
Print *1, Z(I)
Stop
End

Example: Write a computer program to read in the element values of X(12), then add the value of (0) to become the (5th) element in the new array X(13)

Solution:
Dimension X(13)
Do 10 I = 1, 12
Read *1, X(I)
Continue
Read *1, D
Do 15 J = 13, 6, -1
X(J) = X(J - 1)
X(5) = D
X(6) = X(5)
Continue
X(5) = D
Do 20 K = 1, 13
Print *1, X(K)
Stop

Example: Write a program to read in the element values of array $x(15)$, then add the elements of array $y(5)$ to the elements of array $x(5)$ starting from sixth position.

Solution:

```
Dimension X(20), Y(5)
10 DO 10 I = 1, 15
   10 READ *, X(I)
20 DO 20 J = 1, 5
   20 READ *, Y(J)
15 DO 15 K = 6, 15
   15 X(K+5) = X(K)
25 DO 25 L = 6, 10
   25 X(L) = Y(L-5)
30 DO 30 H = 1, 20
   30 PRINT *, X(H)
50 STOP
60 END
```

Example: Write a program to read in the element values of array $x(n)$, then print out the array in descending order.

Solution:

```
READ *, N
Dimension X(N)
10 DO 10 K = 1, N
   10 READ *, X(K)
20 DO 20 I = 1, N-1
   20 DO 30 J = I+1, N
      30 IF (X(J) .LE. X(I)) GOTO 30
   30 A = X(I)
50 STOP
```

```

Example: Write a program to read in the element values of array $x(n)$, then print out the array in descending order.

Solution:

```
READ *, N
Dimension X(N)
10 DO 10 K = 1, N
   10 READ *, X(K)
20 DO 20 I = 1, N-1
   20 DO 30 J = I+1, N
      30 IF (X(J) .LE. X(I)) GOTO 30
   30 A = X(I)
50 STOP
```
Example: Write a program to read in the element values of the array A(10), print out the elements which have values less than ten and greater than zero, also compute and print the summation value of the (-ve) elements.

Solution:

Dimension A(10)
S = 0.0
Do 10 I = 1, 10
Read *, A(I)
IF (A(I) .GE. 0.0 .AND. A(I) .LE. 10.0) Print *, A(I)
IF (A(I) .LT. 0.0) Then
S = S + A(I)
End IF
Continue
Print *, 'Summation of -ve values = ', S
Stop
End
Two-Dimensional Array (Matrix)

Matrix Name (Subscript for rows, Subscript for Columns)

Input statement of 2-Dimensional Arrays

1. Reading the Matrix by Rows

\[
\begin{bmatrix}
X_{11} & X_{12} & X_{13} \\
X_{21} & X_{22} & X_{23}
\end{bmatrix}_{2 \times 3}
\]

Dimension X(2,3)
Do 5  I = 1, 2
Do 7  J = 1, 3
Read \(x, \ X(I,J)\)
7    Continue
5    Continue
Stop
End

OR

Dimension X(2,3)
Do 10  I = 1, 2
Read \(x, (X(I,J), J = 1, 3)\)
10    Continue
Stop
End

OR

Dimension X(2,3)
Read \(x, ((X(I,J), J = 1, 3), J = 1, 2)\)
Stop
End
2. Reading the Matrix by Columns

<table>
<thead>
<tr>
<th>Dimension</th>
<th>X(2,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do 20 J = 1, 3</td>
<td></td>
</tr>
<tr>
<td>Do 25 I = 1, 2</td>
<td></td>
</tr>
<tr>
<td>Read*, X(I,J):</td>
<td></td>
</tr>
<tr>
<td>25 continue</td>
<td></td>
</tr>
<tr>
<td>20 continue</td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td></td>
</tr>
<tr>
<td>End</td>
<td></td>
</tr>
</tbody>
</table>

OR

<table>
<thead>
<tr>
<th>Dimension</th>
<th>X(2,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do 8 J = 1, 3</td>
<td></td>
</tr>
<tr>
<td>Read*, [(X(I,J), I = 1, 2)]</td>
<td></td>
</tr>
<tr>
<td>8 continue</td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td></td>
</tr>
<tr>
<td>End</td>
<td></td>
</tr>
</tbody>
</table>

OR

<table>
<thead>
<tr>
<th>Dimension</th>
<th>X(2,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read*, [(X(I,J), I = 1, 2), J = 1, 3]</td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td></td>
</tr>
<tr>
<td>End</td>
<td></td>
</tr>
</tbody>
</table>
Example: Write a program to read in the matrix $A(2,4)$ and then print out each row in one line.

Solution:

```plaintext
dimension a(2,4)
do 5 i = 1, 2
   do 5 j = 1, 4
      read*, a(i,j)
do 10 i = 1, 2
   print*, (a(i,j), j = 1, 4)
   print*, ' '  
   continue
10  continue
   stop
   end
```

Example: Write a program to read in a matrix $[A]$ which contains 8 rows and 5 columns, then compute and print out the summation of each row alone.

Solution:

```plaintext
dimension a(8,5), sum(8)
do 10 i = 1, 8
   do 10 j = 1, 5
10  read*, a(i,j)
do 15 i = 1, 8
   sum(i) = 0.0
   do 17 j = 1, 5
      sum(i) = sum(i) + a(i,j)
17  continue
15  continue
   do 20 k = 1, 8
      print*, sum(k)
20  continue
   stop
   end
```
Example: Write a program to read in the element values of Matrix $A(6,5)$. Remove the elements of the 3rd row and print out the two main diagonals of the new matrix.

Solution:

```
Dimension A(6,5)
Read *, ((A(I,J), I = 1, 6), J = 1, 5)
Do 15 K = 3, 5
  Do 15 L = 1, 5
  A(K,L) = A(K+1,L)
15 Continue
Do 20 I = 1, 5
  Print *, A(I,I)
20  Do 25 J = 1, 5
25  Print *, A(J, 6-J)
Stop
End
```

Example: Write a program to read in the element values of Matrix $Z(L,K)$ and print the largest three elements in the matrix $[Z]$. 

Solution:

```
Read *, L, K
Dimension Z(L,K), B(L*K)
Do 18 I = 1, L
  Do 18 J = 1, K
18  Read *, Z(I,J)
H = 0
Do 20 I = 1, L
  Do 20 J = 1, K
  H = H + 1
B(H) = Z(I,J)
```
Example: Write a program to compute the value of $D$ given by the equation below, use a subroutine to evaluate the value of the factorials.

$$D = \sum_{i=1}^{n} \frac{(J)! + I^2y}{\sqrt{I!}}$$

Solution:

Read $x$, $n$, $J$, $y$

$D = 0.0$

Call Subroutine $F_1$, $J$

$D = 0$

Do $I = 1, n$

Call Factorial $F_2$, $I$

$D = D + \frac{(F_1 + (I*2)*y)}{\sqrt{F_2}}$

Continue

Print $x$, $D$

Stop

End

Subroutine Factorial $F$, $M$

$F = 1$

Do $K = 1, M$

$F = F * K$

Return

End
Example: Write a program to read in the element values of \( Z(50) \), then calculate the summation of odd and even number of elements in subroutine (summation) then print out the summations in the main program.

Solution:

```
Dimension Z(50)
Read*(Z(I), I = 1, 50)
Call Subroutine (Z, 50, SE)
Print*, 'Sum of odd = ', SO
Print*, 'Sum of even = ', SE
Stop
End

Subroutine summation (Z, 50, SE)
Dimension Z(50)
SO = 0.0
SE = 0.0
Do 10 I = 1, 50, 2
  ZO = SO + Z(I)
10 Continue
Print*, 'Sum of odd = ', SO
Print*, 'Sum of even = ', SE
```

There are two types of files:
1. Formatted files.
2. Unformatted files.

* These files can be used depending on the type of record (Form =):
  Form = Formatted
or
  Form = Unformatted

* state of the file (status =)
  status = new
  status = old
  status = scratch

* Rewind statement
  عجلة الفالد
  تقوم هذه الإملاء بإعادة مساحة الفالد. إذن يتم إعادة الفالد كإعادة تحول الأدوار إلى نقطة الصفر.
  باستثناء الإثبات:

Rewind (unit number)
as:
  Rewind (3)

Open (3, File = 'First year. Data status = 'Old')
(Old
  لسد هذه الإملاء سجميع البرنامج الذي يتم ذلك بروز فلاد Firstyear وotr لسلة امتداد (8)
Open (18, status = 'scratch', Form = 'unformatted')
  لسد هذه الإملاء سجميع الفالد المتصلة بالفترة المطلوبة (8) كدك الدوريات بين استعمال الإثبات.
# Using Files in Programming

To open a file use the statement:

```plaintext
Open (unit number, File = 'Name of the file .Dat') as:
```

For example:

```plaintext
Open (3, File = 'First year .Dat')
```

To close a file use the statement:

```plaintext
Close (unit number)
```

For example:

```plaintext
Close (3)
```
Format (3 F10.1)

\[ P = a + b + c \]
\[ S = P/2.0 \]
\[ Area = \text{SQRT}\left( S \times (S-a) \times (S-b) \times (S-c) \right) \]

Print 12, A, B, C, P, Area

Format (3 (F5.1, 3x)) // 4x // F10.1 // 4x, F12.2
Stop
End

اللاحقة: عند استخدام إصدار مراهنة استمرت كي الرياح كي لم تجعل

ماضية وصفيت في الصناعة.

<table>
<thead>
<tr>
<th>88.6</th>
<th>42.4</th>
<th>56.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7 8 9 10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \Rightarrow \begin{array}{c}
38.6 \\
42.4 \\
56.1
\end{array} \]

Write Statement

رسالة امتحان رسالة طالب سهيلة راجعة بملحة (Print) كان هنا ليتساءل لنسرى تابع تعليق الملك

Write (*, *)

كثير ماء:

Format Statement

<table>
<thead>
<tr>
<th>0.123</th>
<th>0.123</th>
</tr>
</thead>
</table>
The use of this formula is to transform the printed results for a line to another as:

Example: The triangle shown below has the sides given in the figure:
its perimeter is \( P = a + b + c \)
its Area \( \text{Area} = \sqrt{s(s-a)(s-b)(s-c)} \)
where \( s = P/2 = (a+b+c)/2 \)
Compute and print out \( P \) and Area of the triangle using formatted input and output statements.

Solution:
You can also use Hollerith Symbol in this type of statements as:

```
Character $6, C_1, C_2, C_3, C_4
C_1 = 'Red'
C_2 = 'Yellow'
C_3 = 'Blue'
C_4 = 'White'
Print 77, C_1, C_2, C_3, C_4
Format (13 H Colours are =, 4A7)
```

```
Colours are = ☐ ☐ ☐ ☐ Red ☐ ☐ ☐ ☐ yellow ☐ ☐ ☐ ☐ blue
☐ ☐ ☐ ☐ white
```

**Formula X**

The use of this formula is to leave blanks between the printed values.

```
K = 120
A = 1.5
Print 10, A, K
Format (5X, F3.1, 6X, I3)
```

```
☐ ☐ ☐ ☐ ☐ 1.5 ☐ ☐ ☐ ☐ ☐ ☐ ☐ 120
```
b. Formula (E) for real variable with exponent Ew.d or rEw.d

\[ W \geq d + 3 \]

A = 0.5679 E6
X = 0.0057
y = 1.5 E-10
E = -33.2143 E12
Print 70, A, x, y, E
70 Format (E12.4, E10.2, E10.2, E14.6)

\[ \begin{array}{c}
\text{a. 0.5679 E06} \\
\text{b. 0.57 E-02} \\
\text{c. 0.15 E-09} \\
\text{d. 0.33243 E04}
\end{array} \]

Note: In formula (E), (w) must be:

\[ W \geq d + 7 \]

3. Format Statement for Character Variable
A, Aw, rA, rAw

Character Name *10, City *9, Class *7
Name = 'Mohammed'
City = 'Baghdad'
Class = 'First'
Print 10, Name, City, Class
N1 = 15
N2 = 30
N3 = 125
Print 8, N1, N2, N3
Format (2I3, I2)

Format Statement for Real Variables
a. Formula (F): for normal real variable
   F W.d or r F W.d
   d
   real variable index
   9: size of the variable

   A = 1.5
   B = -0.561
   C = 120.25
   Print 7, A, B, C
   Format (F4.1, F7.3, F7.2)

   A = 1.5
   B = 0.561
   C = 120.25
   Print 7, A, B, C
   Format (3F10.3)

   A = 1.5
   B = 0.561
   C = 120.25
Format Statement

Print y, Variable Name
Format statement number

as:
Print 25, A, B, C
25 Format (-------)

1. Format Statement for Integer Variables
   IW or rIW

I: integer variable index
W: size of integer variable
r: repeating index

N1 = 15
N2 = 30
N3 = 125
Print 8, N1, N2, N3
8 Format (I3, I3, I4)

OR

N1 = 15
N2 = 30
N3 = 125
Print 8, N1, N2, N3
8 Format (3I4)

→ □□15 □□30 □□125
□□□□□□□□□□□□□□□□□□□□□□□□□□□

OR

N1 = 15
N2 = 30
N3 = 125
Print 8, N1, N2, N3
8 Format (3I4)

→ □□□15 □□□30 □□□125
Example: Write a program to multiply the matrix $[A]_{m \times n}$ by the matrix $[B]_{n \times L}$ to get the matrix $[D]_{m \times L}$.

Solution:

```plaintext
Read A(m,n), B(n,L), D(m,L)
Read A(i,j), (i = 1, m), (j = 1, n)
Read B(i,j), (i = 1, n), (j = 1, L)
Do 10 II = 1, m
   Do 10 JJ = 1, L
      D(II, JJ) = 0.0
      Do 20 K = 1, n
         D(II, JJ) = D(II, JJ) + A(II, K) * B(K, JJ)
      Continue
   Continue
Do 100 S = 1, m
   Do 100 P = 1, L
      Print *, D(S, P)
   Continue
Step
End
```
Example: Given the matrices $A(20, 10)$, $B(10, 50)$, $D(10, 10)$, we program to calculate the matrix $(R)$ from:

$$R(I, J) = \sum_{K=1}^{10} \frac{A(I, K) \times B(K, J)}{D(K, K)}$$

Solution:

- Dimension $A(20, 10)$, $B(10, 50)$, $D(10, 10)$, $R(20, 50)$
- Read*, $((A(I, J), I = 1, 20), J = 1, 10)$
- Read*, $((B(I, J), I = 1, 10), J = 1, 50)$
- Read*, $((D(I, J), I = 1, 10), J = 1, 10)$
- Do 10 $I = 1, 20$
- Do to $J = 1, 50$
- $R(I, J) = 0.0$
- Do 20 $K = 1, 10$
- $R(I, J) = R(I, J) + \frac{A(I, K) \times B(K, J)}{D(K, K)}$
- Continue
- Continue
- Do 15 $I = 1, 20$
- Do 15 $J = 1, 50$
- Print*, $R(I, J)$
- Stop
- End
Example: Write a program to read in the element values of array A(30) then re-arrange the array $A^2$ as two dimensional matrix $B$, each row contains (6) elements, then print out the new matrix $B$ row by row.

Solution:

```plaintext
Dimension A(30), B(5,6)
Do 10 I = 1, 30
   Read *, A(I)
   Continue
K = 0
Do 12 I = 1, 5
   Do 12 J = 1, 6
      K = K + 1
      B(I, J) = A(K)
   End
Print *, (B(I, J), I = 1, 5, J = 1, 6)
10 Continue
12 Continue
Stop
End
```
Example: Write a program to calculate the element values of $A(8,12)$, where is determined by:

$$A(I, K) = \left( \frac{I+K}{\log (I+K)} \right)^{2/5}$$

Then calculate the element values of matrix $[Z]$ from the for $Z(I, K) = A(I, K) - y(K)$, where $y$ is an array each of its elements is the mean value for one column of matrix $[A]$ in sequence.

Solution:

```
Dimension A(8,12), Z(8,12), y(12)
Do 15 I = 1, 8
Do 16 J = 1, 12
AC(I, K) = (I+K)/(\log (I+K)) \#** (2.0/5.0)
15 Continue
16 Continue
Do 18 J = 1, 12
y(J) = 0.0
Do 20 I = 1, 8
y(J) = y(J) + AC(I, J)
```
Write a program to read in two lists that may be of different size and that then find and display the intersection of the lists, that is the location of items that are common to both lists.

Dimension R(10), W(5)

Read A, (R(I), I=1,10), (W(J), J=1,5)

Do 100 I=1,10
   Do 200 J=1,5
      IF (R(I) .EQ. W(J)) THEN
         Print, R(I), I, J
      END IF
   200 Continue
100 Continue
STOP
END
- Write a computer program for multiplying the elements of the wattage table by the elements of the time table to produce a kilowatt-hour table, indicating power usage for month.

\[ \text{Watt} \times \text{hours} = \text{kilowatt-hours} \]

Real kWh(I)

Dimension watt $\times$ hour

Read (5,70) (watts(I), I=1,12)

Read (5,70) (hours(I), I=1,12)

To format (12 F10.1)

Do 100 I=1,12

\[ \text{kWh}(I) = \frac{\text{watts}(I) \times \text{hours}(I)}{1000} \] for K

Write (6,10) watts(I), hours(I), kWh(I)

To format (1X, F6.1, 5X, F10.1, 4X, I6)

100 Continue

STOP

END
Write a computer program to read in a list of tests, calculate their mean, and then print a list of tests which are greater than mean.

Parameter (N, 10)
Read Sum, Aver,
Read * (test(I), I = 1; N)
Sum = 0.0
Do 20 I = 1, N
   Sum = Sum + test(I)
20 Continue
   Aver = Sum / N
Do 50 I = 1, N
   IF (test(I), GE. Aver) PRINT *, test(I)
50 Continue
End

Given 25 positive integers, write a program which finds the second largest integer number.

Dimension Num(25)

Read *1, (Num(I), I = 1, 25)

Max = Num(1)

Do 10 I = 2, 25

10 IF (Num(I) .GE. Max) Then

   Max = Num(I)
   IMax = I

END IF

10 Continue

Max = 0

Do 30 I = 1, 25

30 IF (I.EQ.IMax) go to 30

IF (Num(I) .GT. Max) Max = Num(I)

30 Continue

Print Max

Stop
END
Write a computer program which sums the element above the main diagonal in a matrix \( A(N \times N) \)

Parameter \((N=5)\)

Dimension \( A(N,N) \)

Read \(* \((A(I,J), I=1,5), J=1,5)\)

Do 10 \( I=1,4 \)
    Do 20 \( J=2,5 \)
    If \((I, LT, J) \) go to 50
    go to 20

50 \( \text{Sum} = \text{Sum} + A(I,J) \)

20 Continue

10 Continue

Print \( ' \text{Sum of the elements above the main diagonal: Sum} \)

Stop
END
parameter (N=5)

Dimension ML(N, N)

Real * , (ML(I, J), I=1, N), J=1, N)

S1=0

S2=0

Do 10 I=1, N

S1 = S1 + ML(I, I)

S2 = S2 + ML(I, 6-I)

c0 Continue

AV1 = S1 / 5.0

AV2 = S2 / 5.0

If (AV1 > AV2) Then

Print *, 'AV1 = ', AV1

Else

Print *, 'AV2 = ', AV2

End If

Stop

END
Parameter \((N=5)\)

Dimension \(K(5, 5)\)

\[
\begin{align*}
\text{Do 10 } & \text{ J = 1, 5 } \\
\text{Do 10 } & \text{ I = 1, 5 } \\
\text{Read } & \text{ #, } K(I, J) \\
\end{align*}
\]

\[10 \text{ Continue}\]

\[
\begin{align*}
\text{Do 100 } & \text{ J = 2, 4 } \\
\text{Save } & = K(1, J) \\
K(1, J) & = K(5, J) \\
K(5, J) & = \text{ Save} \\
\text{100 Continue}\end{align*}
\]

\[
\begin{align*}
\text{Do 200 } & \text{ I = 2, 4 } \\
\text{Save } & = K(I, 1) \\
K(I, 1) & = K(I, 5) \\
K(I, 5) & = \text{ Save} \\
\text{200 Continue}
\end{align*}
\]
\[ \text{Save} = k(2, 3) \]
\[ k(2, 3) = k(4, 3) \]
\[ k(4, 3) = \text{Save} \]
\[ \text{Save} = k(3, 2) \]
\[ k(3, 2) = k(3, 4) \]
\[ k(3, 4) = \text{Save} \]

Do 100 I = 1, 5
Do 200 J = 1, 5
print \&, k(I, J)
200 Continue
100 Continue
Stop
END
العديد من القياسات المعلومة (10,20) 

• تتبع جميع القياسات المعلومة ووضع القيم المذكورة في نقاط المعلومة (10,20) وكذلك ضمن ineffective. يتم تكرار كل جزء من المعلومة (10,20) ووضع القيم المذكورة في نقاط المعلومة (10,20).

Dimension A(10,20), S1(10), S2(20)
Read (5,10) (A(I,J), J=1,20, I=1,10)
10 Format (10F5.2)

Do 200 I=1,10
   S1(I)=0.0
   Do 100 J=1,20
100   S1(I)=S1(I)+A(I,J)
200 Continue
   Do 400 J=1,20
      S2(J)=0.0
      Do 300 I=1,10
300   S2(J)=S2(J)+A(I,J)
400 Continue
   Write (6,500) (I, S1(I), I=1,10)
500 Format (5X,'S1(I),'=,F5.2
   Write (6,600) (J, S2(J), J=1,20
600 Format (5X,'S2(J)'=,F7.2
STOP
END
Dimension A(10), B(2,5)
Read, (B(I,J), I=1,2, J=1,5)
Do 10 I = 1, 2
   Do 20 J = 1, 5
      M = M + 1
      A(M) = B(I,J)
   20 Continue
  10 Continue
Do 30 I = 1, 10
   Read
   Print *, A(I)
30 Continue
Stop
END