Assignment-3

Sourav Das (21021085@uopca.unipune.ac.in)

(a) Discuss the validity of the following groups of characters as FORTRAN variables. If you think they are not valid, give reasons.

(i) `STOP`
Ans. This is **valid** as variable.

(ii) `abc-1`
Ans. This is **not** a valid variable, since hyphen/negative sign is not allowed in variables. It is used as subtraction operator.

(iii) `1A2B3C`
Ans. This is **not** a valid variable, since variables cannot start with numbers.

(iv) `do I 3`
Ans. This is also **not** a valid variable, since a variable name cannot have spaces between the characters.

(v) `FUNCTION`
Ans. This is **valid** as a variable.

(b) Discuss the validity of the following statements as FORTRAN statements. If you think they are not valid, give reasons.

(i) `doI = 2.54`
Ans. This is a **valid** statement. `doI` is assigned to the value `2.54`.

(ii) `STOP = END`
Ans. If the type of `STOP` and `END` are declared, then `STOP` and `END` will work as variables instead of FORTRAN keywords, and this statement will assign the value of `STOP` variable to that of `END`, which is a valid statement.
Otherwise, a keyword cannot be assigned to something else. In that case, this statment is invalid.

(iii) `WRITE(I, J) = I + J`
Ans. If the type of `I, J` are declared, and assigned to a value, and `WRITE` is defined as a 2D array, then this statement will perfectly work fine. A minimum working example is given below:

```
1 program test
2 integer, dimension(2, 2) :: WRITE
3 I = 1
4 J = 1
5 WRITE(I, J) = I + J
6 write(*,*) WRITE
7 end program test
```

Here, 2 is assigned to the element which is present in `WRITE(1, 1)`. In line 6, the first `write(*,*)` is acting as the keyword, and the second as variable, although FORTRAN is case-insensitive. So the given statement is **valid** in this case.

Otherwise, if `WRITE` is not declared in the program as a variable, then it will work as the keyword everywhere in the program, and it cannot be assigned to anything else. In that case, this statement will be **invalid**.

(iv) `A + B = C + D`

Ans. This is an **invalid** statement. Here, `=` is acting as an assignment operator. On the left of `=`, a single variable can be used, and on the right, an other variable, or a value, or an expression. Here, two variables are present to the left of `=`, which is not a valid syntax.

(v) `if (I = J) exit`

Ans. This statement is **invalid**. In the parenthesis beside `if`, a *conditional* have to be used, resulting `.true.` or `.false.`. But here, assignment operation is done, which will result in syntax error. To correct this, `==` can be used for comparison of equality of `I` and `J`.

(c) Evaluate the following arithmetic expressions as Fortran arithmetic expressions (use default type for variables) given I = 6, J = 2, K = 3, L = 9, A = 2.4, B = -3.2, C = 2.

(i) `N = I/J + K/(J + 1)`

Ans.

$$N = \frac{6}{2} + \frac{3}{(2+1)}$$
$$N = 3 + \frac{3}{3}$$
$$N = 3 + 1$$
$$\therefore \boxed{N = 4}$$

(ii) `N = I**J**K`

Ans.

$$N = 6^{2^3}$$
$$N = 6^8$$
$$\therefore \boxed{N = 1679616}$$

(iii) `D = C/I + A/K`

Ans.

$$D = \frac{2.0}{6} + \frac{2.4}{3}$$
$$D = 0.33\bar{3} + 0.8$$
$$\therefore \boxed{D = 1.13\bar{3}}$$

(iv) `E = B**A + K/L`

Ans.

$$E = -3.2^{2.4} + \frac{3}{9}$$
$$E = \texttt{NaN} + 0$$
$$\therefore \boxed{E = \texttt{NaN}}$$

Here, $-3.2^{2.4}$ is a complex number. But since, here, `E` is `implicit` data type, i.e., real, so it can not store complex data type. Hence, Not a Number (`NaN`) is shown.

(v) `N = I+J/A*B + C/(J+K)`

Ans.

$$N = 6 + \frac{2}{2.4}(-3.2) + \frac{2.0}{2+3}$$
$$N = 6 - 2.6\bar{6} + 0.4$$
$$N = 3.7\bar{3}$$

But since the implicit data type of N is integer, so, the values after decimal places will be truncated.

$$\therefore \boxed{N = 3}$$

(d) Give the output of the following program segments (use default type for variables).
(i)

```
do I = 1,3
    do J = 1,3
        A(I,J) = I/J *(J/I)
    enddo
    write(*,10) (A(I,K), K = 1,3)
enddo
10 format (5 I 3)
```

Ans.

```
    1   0   0
    0   1   0
    0   0   1
```

(ii)

```
N = 0
do I = 1,3
    do J = 1,I,2
        do K = 1,J
            do L = 1,K
                N = N + 2
            enddo
        enddo
    enddo
enddo
write(*,10) 'The value of N is', N
10   format(I5)
```

Ans. Considering `A17` is also included in the format descriptor before `I5`, the output is:

```
    The value of N is    18
```

(e) Write a program to read the following matrix column-wise and write it as a matrix as it is.
A(1,1) = 1, A(1,2) = 2, A(1,3) = 3
A(2,1) = 4, A(2,2) = 5, A(2,3) = 6
A(3,1) = 7, A(3,2) = 8, A(3,3) = 9
Ans.

```
program MATRIX_COLUMNWISE
    integer :: A(10,10)
    ! A(1,1) = 1; A(1,2) = 2; A(1,3) = 3
    ! A(2,1) = 4; A(2,2) = 5; A(2,3) = 6
    ! A(3,1) = 7; A(3,2) = 8; A(3,3) = 9

    ! Reading the matrix column-wise
    ! row index I will change more than column index J
    write(*,*) "Enter Order of Square Matrix."
    read(*,*) N
    write(*,*) "Enter Matrix Elements column-wise:"
    do J = 1, N
        do I = 1, N
            read(*,*) A(I, J)
        end do
    end do
    write(*,*) "The matrix is:"
    do I = 1, N
        write(*,*) (A(I, J), J=1,N)
```

```
    end do
end program MATRIX_COLUMNWISE
! OUTPUT
! Enter Order of Square Matrix.
! 3
!  Enter Matrix Elements column-wise:
! 1
! 4
! 7
! 2
! 5
! 8
! 3
! 6
! 9
!  The matrix is:
!           1           2           3
!           4           5           6
!           7           8           9
```