

3.1.- EXPRESIONES ARITMETICAS Y DE ASIGNACION

Para poder trabajar con las expresiones aritméticas más comunes, es preciso conocer adecuadamente los *operadores aritméticos* y su nomenclatura. En lenguaje FORTRAN dichos operadores están formados por una serie de caracteres especiales que difieren muy poco de los que estamos acostumbrados a utilizar cuando empleamos una calculadora o el teclado de un ordenador. Los operadores aritméticos reconocidos por el FORTRAN son:

OPERADOR	OPERACION
**	Exponenciación
/	División
*	Multipliación
-	Resta o negación
+	Suma

Una vez conocidos los operadores aritméticos, podremos construir una *expresión aritmética*, la cual estará formada por operandos, es decir constantes y/o variables, operadores y paréntesis de manera que agrupados indiquen la ejecución de un determinado cálculo aritmético. Como ejemplos sencillos de expresiones aritméticas con un sólo operador tendremos:

OPERACION	FORMA ALGEBRAICA	EN FORTRAN
SUMA	$A + B$	$A + B$
RESTA	$A - B$	$A - B$
MULTIPLICACION	$A \times B$	$A * B$
DIVISION	A / B	A / B
EXPONENCIACION	A^n	$A ** n$
NEGACION	$-A$	$-A$

Por otro lado cuando el número de operandos y operadores aumenta las expresiones se complican:

FORMA ALGEBRAICA	EN FORTRAN
$\frac{A+B}{2}$	$(A + B)/2$
$\frac{At^2}{2}$	$0.5*(A*t**2)$
$B^2 - 4AC$	$B**2 - 4*A*C$
$\frac{3Z}{X-Y}$	$3*Z/(X-Y)$
$A(C - D)$	$A*(C - D)$

Existen dos aspectos fundamentales a la hora de escribir una expresión aritmética:

1.- Es preciso tener presente en todo momento los tipos de variables y constantes de la expresión aritmética para conocer el resultado de la operación, el cual puede ser real o entero.

A continuación se muestra una tabla en la que se especifican los resultados para cada una de las operaciones posibles.

+, -, *, /	ENTERO	REAL	D. PRECISION
ENTERO	ENTERO	REAL	D. PRECISION
REAL	REAL	REAL	D. PRECISION
D. PRECISION	D. PRECISION	D. PRECISION	D. PRECISION

En cuanto a la exponenciación se refiere:

		**		
		EXPONENTE		
BASE	ENTERO	ENTERO	REAL	D. PRECISION
	REAL	REAL	REAL	D. PRECISION
	D. PRECISION	D. PRECISION	D. PRECISION	D. PRECISION

2.- Realizar una correcta escritura de la expresión poniendo especial cuidado en la ordenación y criterios de prioridad operacional para lograr la mayor simplificación de la expresión. En este sentido es necesario observar los siguientes ejemplos:

FORMA ALGEBRAICA	EN FORTRAN
$\frac{2}{5(A+B)}$	2 / (5*(A + B))
$\frac{A * B}{C^d - E}$	A * B / (C **d - E)

En el primero de los casos ha sido preciso el empleo de dos niveles de apertura de paréntesis con el fin de salvaguardar la operación completa del denominador; sin embargo en el segundo ejemplo sorprendentemente no han sido utilizados paréntesis. Estas cuestiones responden a una serie de reglas que FORTRAN posee a la hora de escribir una expresión aritmética.

Regla nº 1 : Dos operadores aritméticos nunca pueden estar juntos dentro de una expresión, deberán estar separados por un paréntesis.

<i>Ejemplo:</i>	<u>EXPRESION INCORRECTA</u>	<u>EXPRESION CORRECTA</u>
	A-+B	A-(+B)
	A*-B	A*(-B)
	A--B	A-(-B)
	A**-B	A**(-B)

Regla nº 2 : Por cada paréntesis abierto deberá existir uno de cierre

<i>Ejemplo:</i>	<u>EXPRESION INCORRECTA</u>	<u>EXPRESION CORRECTA</u>
	(A-(+B)	A-(+B)
	(A*(B-C)	(A*(B-C))

Regla nº 3 : Cuando no se hace uso de paréntesis en una expresión FORTRAN por no ser imprescindibles es preciso conocer y aplicar la prioridad operacional establecida para este lenguaje la cual es:

- 1.- POTENCIACION
- 2.- MULTIPLICACION Y DIVISION

3.- SUMA Y RESTA

Cuando en una misma expresión existen operadores de la misma prioridad (multiplicaciones y divisiones o sumas y restas), las operaciones se realizarán recorriendo la expresión *de izquierda a derecha*; para el caso de la potenciación se realizarán *de derecha a izquierda*.

Ejemplos: $A*B / C**D - E$

$A*B / C**D - E$	OPERACION
$C**D$	**
$A*B$	*
$A*B / C**D$	/
$A*B / C**D - E$	-

$A+B+ C+D+ E$

$A+B+ C+D+ E$	OPERACION
$A+B$	+
$A+B+ C$	+
$A+B+C+D$	+
$A+B+C+D+ E$	+

Una vez conocidas las normas básicas para la construcción de las expresiones aritméticas, queda por conocer como asignar el resultado de la expresión a una variable, o como asignar un valor determinado a una variable. Para ello se emplea la *sentencia de asignación aritmética*, la cual debe considerarse una sentencia ejecutable. En FORTRAN existen diferentes sentencias de asignación atendiendo al tipo de dato que se desea asignar.

- **SENTENCIA DE ASIGNACION ARITMETICA**

$$X = E_a$$

Siendo **X** el nombre de una variable de tipo entera, real o doble precisión y E_a una expresión aritmética FORTRAN

- **SENTENCIA DE ASIGNACION COMPLEJA**

$$X = \text{CMPLX} (2.0 , 5.1)$$

En este caso **X** es el nombre de una variable compleja a la cual se le ha asignado el valor del número complejo 2.0+5.1i.

- **SENTENCIA DE ASIGNACION LOGICA**

$$X = E_l$$

En este caso **X** es el nombre de una variable declarada como lógica y E_l es una expresión lógica. (Ejemplo: X=.TRUE.)

- **SENTENCIA DE ASIGNACION CARACTER**

$$X = E_l$$

En este caso **X** es el nombre de una variable declarada como carácter y **E_c** es una expresión carácter.

(Ejemplo: X= 'NOTAS' , **IMPORTANTE**: NOTAS tiene 5 caracteres, luego X debería estar declarada como CHARACTER*5)

EJERCICIO 3.1

1. Programar la siguiente expresión:

$$VALOR = \left(\frac{X^2 + Y^3(X - 3)}{\frac{1}{X} + \frac{Y}{X^2}} \right)^{0.5}$$

El programa deberá leer por pantalla los valores de X e Y, para presentar por pantalla también el valor resultante. Declarar X,Y y VALOR como reales.

2. ¿ Existe algún valor X y/o Y que pueda provocar algún problema al programa al ser ejecutado?. En caso de existir evitar la posible entrada de esos valores.

EJERCICIO 3.2

1. Realizar un programa que al ser ejecutado presente por pantalla el mensaje:
BUENAS TARDES, ESTO ES UNA PRACTICA DE FORTRAN

3.2.- FUNCIONES INTRINSECAS

Las funciones intrínsecas constituyen un conjunto de funciones trigonométricas, exponenciales, logarítmicas y otras muchas que el lenguaje FORTRAN pone a disposición del programador para que las utilice en cualquier momento.

Cada función intrínseca posee un *nombre*, un *tipo* y debe contener uno o varios *argumentos* para su ejecución. Si tomamos como ejemplo la función seno, ésta se define del siguiente modo:

SIN (7.4) calculara el seno de 7.4 radianes

Igualmente esta función puede estar incluida en una expresión aritmética con asignación del siguiente modo:

$X=Y+SIN(7.4)$

La forma general de una función intrínseca es:

NOMBRE (a₁, a₂, a_n)

Siendo **NOMBRE**, la denominación de la función intrínseca preestablecida por el sistema y **a₁, a₂, a_n** los argumentos que cada función precisa para su ejecución.

Las funciones intrínsecas pueden estar anidadas de tal modo que en la siguiente expresión:

$X = ABS (TAN(Z))$

se realizaría primero el cálculo de la tangente de Z y a su resultado se le aplicaría el valor absoluto, para luego ser asignado a la variable X.

A continuación se presenta una tabla con las diferentes funciones intrínsecas presentes en FORTRAN. A la hora de consultar dicha tabla deberá entenderse las siguientes claves:

NOMBRE GENERICO: Nombre de la función intrínseca independientemente del tipo de argumentos que sean utilizados.

NOMBRE ESPECIFICO: Nombre de la función intrínseca específico para el tipo de argumentos empleados.

ARGTS: Tipo y número de argumentos requeridos.

FUNC: Tipo de resultado.

DEFINICION: Explicación de la operación realizada por la función.

NOMBRE GENÉRICO	NOMBRE ESPECIFICO	TIPO		DEFINICIÓN
		ARGTS	FUNC	
INT	- INT IFIX IDINT -	I R R DP C	I I I I I	Calcula la parte entera de un argumento, truncándolo si fuera necesario. Ejemplos: INT(5) = 5 INT(3.5) = 3 INT(3.8,-2.3) =3 (argumento complejo) INT equivale a IFIX
REAL	REAL F L O A T - S N G L -	I I R DP C	R R R R R	Calcula el valor real equivalente al valor del argumento. Ejemplos: REAL(8.2) = 8.2 REAL(2.1, 5.7) = 2.1 REAL(3) =3.0 FLOAT equivale a REAL
DBLE	- - - -	I R DP C	DP DP DP DP	Calcula el valor doble precisión equivalente al valor del argumento. Ejemplos: DBLE(3.ODO) = 3.ODO DBLE(5) = 5.ODO DBLE(7.2,3.5) = 7.2DO
CMPLX	- - - -	1 ó 2 I 1 ó 2 R 1 ó 2DP 1 ó 2 C	C C C C	Calcula el valor complejo equivalente al valor de argumento. Ejemplos : CMPLX(2. 1, 0.7) = (2.1, 0.7) CMPLX(5) = (5.0, 0.0) CMPLX(0.7) = (0.7, 0.0)
	ICHAR	CH	I	El argumento está constituido por un carácter único y la función determina la posición de ese carácter dentro del conjunto de caracteres alfanuméricos ordenados según una determinada secuencia que cada máquina posee. Ejemplo: ICHAR (*) = 5 significa que el asterisco ocupa la posición 6 dentro de la sucesión de caracteres de una máquina conforme al sistema de codificación ASCII. La sucesión comienza en cero.
	CHAR	I	CH	Obtiene como resultado el carácter que ocupa el lugar indicado por el argumento dentro de la secuencia ordenada de caracteres que posee cada máquina y que comienza en la posición cero. Ejemplos: CHAR(3) = (, CHAR(6) = + , CHAR(22) =A
AINT	AINT DINT	R DP	R DP	Producen un truncamiento del argumento x seguido de una conversión real o doble precisión del resultado obtenido, es decir, estas funciones equivalen a REAL (INT(X)) y DBLE (INT(X)). Ejemplos: AINT(5.7) =5.0, DINT(8.3) =8.ODO
ANINT	ANINT DNINT	R DP	R DP	Obtención del entero más próximo (expresado en forma real) al valor del argumento (redondeo). Si es $x > 0$, se calcula REAL(INT(X +0.5)) ó DBLE (INT(X +0.5)) Si es $x < 0$, se calcula REAL (INT(X-0.5)) ó DBLE (INT(X-0.51)) Ejemplos: ANINT(4.3) =4.0 ANINT(-2.4DO) =-2.ODO ANINT(5.6) =6.0
NINT	NINT IDNINT	R DP	I I	Obtención del valor entero más próximo al valor del argumento (redondeo). Si es $x > 0$, se calcula INT (X + 0.5) Si es $x < 0$, se calcula INT (X -0.5) Ejemplos: NINT (10.3) =10, NINT (4.5 DO) =5
ABS	IABS ABS DABS CABS	I R DP C	I R DP R	Calcula el valor absoluto o módulo del argumento, es decir, si xr y xi son las partes reales e imaginarias del argumentos, se calcula: $(xr^2 + xi^2)^{0.5}$ Ejemplos: ABS (-5) = 5 , DABS (2.ODO) = 2.ODO , CABS (4.0, 3.0) = 5.0
MOD	MOD AMOD DMOD	2,I 2,R 2,DP	I R DP	Calcula el resto de dividir el primer argumento X1 por el segundo X2, es decir, siendo $x2 \neq 0$, Ejemplos: MOD(X1, X2) = X1 - (INT (X1 /X2)*X2) AMOD(X1, X2) = REAL (X1 - (INT (X1 /X2) * X2) DMOD (X1, X2) = DBLE (X1 - (INT(X1/X2) * X2
SIGN	ISIGN SIGN DSING	2,I 2,R 2,DP	I R DP	Transferencia de signo de un argumento X1 al otro X2, es decir: Si es $X2 > 0$, se calcula ABS(X1) Si es $X2 < 0$, se Calcula -ABS(X1)
DIM	IDIM DIM DDIM	2,I 2,R 2,DP	I R DP	Calcula la diferencia positiva entre los dos argumentos X1 y X2, es decir X1 -X2, Si es $X1 > X2$ o cero en otros casos.
MAX	MAXO AMAX1 DMAX1	2,I 2,R 2,DP	I R DP	Calcula el máximo de un conjunto de valores o argumentos. Ejemplos: AMAXO (3,-7, 4, -8) =4.0 ,MAX (2.5, 6.2, - 7, 1) = 6.2
	AMAXO MAX1	2,I 2,R	R I	

MIN	MIN0 AMIN1 DMIN1	2,I 2,R 2,DP	I R DP	Calcula el mínimo de un conjunto de valores o argumentos. Ejemplos: MIN1 (-1.5, 2.8, 03)=-1 MIN0 (4, 3, 2, -7, -9) = -9
	AMIN0 MIN1	2,I 2,R	R I	
	DPROD	2,R	DP	Dados dos argumentos reales, calcula su producto expresándole en doble precisión
	LEN	CH	I	Obtiene la longitud de una constante <i>carácter</i> . Ejemplos: LEN('AMIGO') =5, LEN ('AL_SOL') =6
	INDEX	2,CH	I	INDEX (X1, X2). Obtiene la posición de la primera aparición de la serie de caracteres X2 en la serie de caracteres X, . Si la serie X2 no aparece en la X se devuelve el valor cero. Ejemplo: si fuera AX2 ='AMANECER_CLARO' entonces, sería INDEX (AX2,'CLA') = 10
	AIMAG	C	C	Obtiene la parte imaginaria de un argumento complejo. Ejemplo: AI MAG (3.2, -7.5) = -7.5
	CONJG	C	C	Obtiene el complejo conjugado de otro. Ejemplos: CONJG (3.2, 2.7) = (3.2, -2.7) CONJG(1.5, 0.0) = (1.5, 0.0)
SQRT	SQRT DSORT CSQRT	R DP C	R DP C	Calcula la raíz cuadrada del argumento que tiene que ser positivo.
EXP	EXP DEXP CEXP	R DP C	R DP C	Calcula el valor de <i>e</i> elevado a la potencia indicada por el argumento, siendo <i>e</i> la base de los logaritmos naturales o neperianos.
LOG	ALOG DLOG CLOG	R DP C	R DP C	Calcula el logaritmo en base e o logaritmo natural del valor del argumento.
LOG10	ALOG10 DLOG10	R DP	R DP	Calcula el logaritmo en base 10 ó logaritmo común del valor del argumento.
SIN	SIN DSIN CSIN	R DP C	R DP C	Calcula el seno del argumento que debe expresarse en radianes. Una circunferencia tiene 2pi radianes.
COS	COS DCOS CCOS	R DP C	R DP C	Calcula el coseno del argumento que debe expresarse en radianes.
TAN	TAN DTAN	R DP	R DP	Calcula la tangente trigonométrica del argumento que debe expresarse en radianes.
ASIN	ASIN DASIN	R DP	R DP	Calcula la función arco seno del argumento, es decir, el ángulo en radianes que corresponde a un argumento seno.
ACOS	ACOS DACOS	R DP	R DP	Calcula la función arco coseno del argumento, es decir, el ángulo en radianes que corresponde a un argumento coseno.
ATAN	ATAN DATAN	R DP	R DP	Calcula la función arco tangente del argumento, es decir, el ángulo en radianes que corresponde a un argumento tangente.
ATAN2	ATAN2 DATAN2	2,R 2,DP	R DP	Calcula la función arco tangente del cociente de los dos argumentos, es decir, el ángulo en radianes que corresponde a la tangente trigonométrica cociente de ambos argumentos.
SINH	SINH DSINH	R DP	R DP	Calcula el valor del seno hiperbólico del argumento, es decir, $(e^x - e^{-x}) / 2$, siendo <i>x</i> el argumento.
COSH	COSH DCOSH	R DP	R DP	Calcula el valor del coseno hiperbólico del argumento, es decir, $(e^x + e^{-x}) / 2$, siendo <i>x</i> el argumento.
TANH	TANH DTANH	R DP	R DP	Calcula el valor de la tangente hiperbólica del argumento, es decir, $(e^x - e^{-x}) / (e^x + e^{-x})$, siendo <i>x</i> el argumento.
	LGE	2,CH	L	Siendo X1 y X2 los dos argumentos, esta función devuelve el valor .TRUE. si es X1=X2 o bien si X1 precede a X2 en la sucesión de caracteres ordenada según la norma ASCII. En caso contrario devuelve el valor .FALSE.
	LGT	2,CH	L	Siendo X1 y X2 los dos argumentos, esta función devuelve el valor -TRUE. si X1 sigue a X2 en la sucesión de caracteres ordenada según la norma ASCII. En caso contrario devuelve el valor -FALSE
	LLE	2,CH	L	Siendo X1 y X2 los dos argumentos, esta función devuelve el valor -TRUE. si es X1=X2 o bien X1 precede a X2 en la sucesión de caracteres ordenada según la norma ASCII. En caso contrario devuelve el valor .FALSE.
	LLT	2,CH	L	Siendo X1 y X2 los dos argumentos, esta función devuelve el valor .TRUE. si X1 precede a X2 en la sucesión de caracteres ordenada según la norma ASCII. En caso contrario devuelve el valor .FALSE.

ABREVIATURAS: I = ENTERO, R = REAL, CH = CARACTER, C = COMPLEJO, L= LOGICA, DP = DOBLE PRECISION

EJERCICIO 3.3

1. Realizar un programa cuya única entrada sea un número real por pantalla y su respuesta sean dos números distintos, uno la parte entera y otra la parte decimal del número introducido.

EJERCICIO 3.4

1. Realizar un programa que calcule el máximo y el mínimo de un conjunto de 5 valores leídos por pantalla. El resultado deberá ser escrito en pantalla.

EJERCICIO 3.5

1. Programar la siguiente expresión aritmética:

$$TOTAL = \left(\frac{\sin(X^2) + \sqrt{Y^3(X^2 + 3)}}{\frac{|X - Y|}{\log_{10} X}} \right) (3.0 + 6.7j)$$

El programa deberá leer por pantalla los valores de X e Y, para presentar por pantalla también el valor resultante.

En la pantalla resultante deberán aparecer los resultados del siguiente modo:

```
VALOR DE X = X
VALOR DE Y = Y
RESULTADO = TOTAL
```

EJERCICIO 3.6

1. Desarrollar un programa que tenga como entradas cuatro números enteros: a, b, c y d, correspondiéndose con los dos números complejos, A=a+bi y B=c+di. El programa deberá calcular A+B, A-B, A*B y A/B, y presentar por pantalla los resultados de manera comprensible.

EJERCICIO 3.7

1. Desarrollar un programa que calcule la distancia entre dos puntos en R²