

# Computación - Práctica 2

## Introducción a la programación en FORTRAN

Quienes fuera de la sala informática quieran realizar los ejercicios de la práctica y no tengan acceso a computadoras con sistema operativo Linux con el compilador *gfortran* instalado tienen varias opciones, sin necesidad de cambiar de sistema operativo.

Si poseen computadoras con sistema operativo Windows pueden descargar e instalar un compilador FORTRAN para Windows. Alternativamente, quienes posean Windows 10 o posteriores pueden instalar el Ubuntu para Windows y allí instalar el compilador *gfortran*. En la página de la cátedra, hay un apunte con las instrucciones para instalarlo:

<http://computacion.sytes.net/PDFs/WSL.pdf>.

Quienes no puedan instalar un compilador FORTRAN o no posean una computadora pueden utilizar el compilador FORTRAN on-line que se encuentra en la página web:

[https://www.onlinegdb.com/online\\_fortran\\_compiler](https://www.onlinegdb.com/online_fortran_compiler)

Este compilador on-line puede utilizarse también en celulares y tablets.

En los enunciados de las prácticas, las instrucciones de compilación corresponden al compilador *gfortran*. Quienes utilicen otros compiladores deberán utilizar las instrucciones propias de cada compilador.

Si usan el compilador FORTRAN online, los programas se compilan y ejecutan presionando el botón Run. **Nota:** Si se usa el compilador on-line, en los programas descargados de la página deben cambiar la letra C, utilizada para representar un comentario en FORTRAN 77 por el signo de admiración ! (utilizado en FORTRAN 90 y posteriores).

**Ejercicio 1.** Bajar de la página de la cátedra el programa *Volumen.f*, que permite calcular el volumen de una esfera cualquiera a partir de su radio.

- Identificar las distintas partes del programa en lenguaje FORTRAN.
- Compilar (`gfortran -Wall Volumen.f`).
- Correr el programa ejecutable (`./a.out`) y verificar que da un resultado correcto.

**Ejercicio 2.** Utilizar un editor de textos para almacenar el código que se detalla abajo en el archivo fuente *Calendario.f*.

```
PROGRAM CALEND
WRITE(*,*) "14 de julio de 1789"
END
```

- Compilar.

- b) Ejecutar el programa.
- c) Modificar el programa para que anuncie la fecha actual. Compilar esta versión del programa para que el ejecutable se llame `Calendario` (`gfortran -Wall -o Calendario Calendario.f`). Corroborar que el programa da el resultado correcto.

**Ejercicio 3.** Bajar de la página de la cátedra el programa *Triangulo.f*, que permite calcular el área de un triángulo cualquiera a partir de sus lados  $l_1$ ,  $l_2$  y  $l_3$ , de acuerdo a la fórmula:

$$A = \sqrt{s(s-l_1)(s-l_2)(s-l_3)}, \quad \text{donde } s = \frac{1}{2}(l_1 + l_2 + l_3). \quad (1)$$

- a) Identificar las distintas partes del programa en lenguaje FORTRAN.
- b) Compilar.
- c) Si fueron identificados errores, corregirlos y repetir el procedimiento desde b).

**Ejercicio 4.** Compilar la versión corregida del programa. Correr un ejemplo con lados de longitud 1. ¿Es correcto el resultado?

**Ejercicio 5.** Compilar la versión corregida del programa del ejercicio anterior, para que el ejecutable se llame *Triangulo*. Controlar los resultados usando distintas ternas de prueba.

**Ejercicio 6.** Descargar el archivo *Detectando-Errores.f* de la página de la Cátedra. Es un programa de asignación de valores, sin una función específica, sólo ilustrativo.

Compilar el programa con:

```
> gfortran Detectando-Errores.f
```

y usando

```
> gfortran -Wall Detectando-Errores.f
```

¿Observa diferencia entre los mensajes de error en cada compilación? Interprete los mensajes de error. ¿Cómo los solucionaría?

Sobre el programa sin modificar (descargarlo nuevamente si es necesario), ejecutar:

```
> ftnchek Detectando-Errores.f
```

¿Los mensajes de error son los mismos que en los casos anteriores?

**Nota:** *ftnchek* no compila el programa, sólo lo analiza y brinda información adicional sobre errores que el compilador no detecta (sólo para FORTRAN 77).

Los siguientes ejercicios (7 al 11 inclusive) deben realizarse previamente sin programar. Luego para verificar si las conclusiones obtenidas son correctas, comparar con los resultados obtenidos programando los distintos ítems.

**Ejercicio 7.** Determinar cuáles de los siguientes nombres de variables son correctos para FORTRAN.

- a) `aso` b) `Veo` c) `kk-1` d) `kk_1`  
 e) `1t4` f) `s41#` g) `st uv` h) `computacion`

**Ejercicio 8.** Determinar el valor de la variable real  $A$  o de la variable entera  $I$  obtenido como resultado de cada una de las siguientes proposiciones de asignación.

- a)  $A = 2 * 6 + 1$  l)  $A = 10.0 ** (-4)$   
 b)  $A = 2 / 3$  m)  $A = 1./3. + 1./3. + 1./3.$   
 c)  $A = 2.* 6./ 4.$  n)  $A = 1/3 + 1./3.+ 1/3$   
 d)  $I = 2 * 10 / 4$  ñ)  $A = 4.0 ** (3 / 2)$   
 e)  $I = 2 * (10 / 4)$  o)  $A = 4.0 ** 3./ 2.$   
 f)  $A = 2 * (10 / 4)$  p)  $A = 4.0 ** (3./ 2.)$   
 g)  $A = 2.* (10./ 4.)$  q)  $I = 19 / 4 + 5 / 4$   
 h)  $A = 2.0 * (1.0E1 / 4.0)$  r)  $A = 19 / 4 + 5 / 4$   
 i)  $A = 6.0 * 1.0 / 6.0$  s)  $I = 100 * (99 / 100)$   
 j)  $A = 6.0 * (1.0 / 6.0)$  t)  $I = 10 ** (2 / 3)$   
 k)  $A = 10 ** (-4)$  u)  $I = 10 ** (2./ 3.)$

**Ejercicio 9.** Determinar si las siguientes expresiones están mal, o inconvenientemente escritas en FORTRAN. ¿Por qué?

- a)  $a(b+c)$  `A(B+C)` j)  $t = 3 \times 10^{-6}$  `T=3*10**-6`  
 b)  $2a+4$  `2*A+4` k)  $t = 3 \times 10^{-6}$  `T=3*10**(-6)`  
 c)  $a^{n+1}$  `A**N+1` l)  $\frac{xy}{z+1}$  `XY / Z+1`  
 d)  $a^{(1/n)}$  `A**(1/N)` m)  $\cos^{-1}(|\ln(x)|)$  `ACOS(LOG(ABS(X)))`  
 e)  $\frac{ab}{cd}$  `A*B/C*D` n)  $x^a x^b$  `X**A**B`  
 f)  $(-x)^n$  `-X**N` ñ)  $(x^a)^b$  `X**A**B`  
 g)  $t = 3 \times 10^6$  `T=3.*10.**6` o)  $\log|ab|$  `LOG(ABS(A*B))`  
 h)  $t = 3 \times 10^6$  `T=3.0e6` p)  $e^{|a|} - \frac{b^2}{|c|}$  `EXP(ABS(A)-B**2/ABS(C))`  
 i)  $t = 3 \times 10^{-6}$  `T=3.0e-6`

**Ejercicio 10.** Dadas las variables con los valores que se indican:  $A = 2$ ;  $B = 5$ ;  $C = 10$ ;  $D = 2.5$ ;  $E = -4$ ;  $I = 2$ ;  $J = 3$ ;  $K = -2$ ;  $F = .FALSE.$ ;  $G=1$ ;  $T = .TRUE.$ ;  $Z=-1$ , deducir el valor lógico de cada una de las siguientes expresiones lógicas:

- a) `T.AND.F.OR..FALSE.`  
 b) `A**I+B.LE.B/C+D`  
 c) `I/J.EQ.2+K.AND.B/C+D.GE.E+C/D-A**J`  
 d) `C**2+D*E-10.0E+2.LE.B+2.0*C.AND.(.FALSE..OR..NOT..FALSE.)`  
 e) `(B*J+3.0).EQ.(D-E).AND.(.NOT.F)`

f)  $((2+A) \cdot LT \cdot B) \cdot OR \cdot (C \cdot GE \cdot J \cdot AND \cdot (5 \cdot G) \cdot NE \cdot Z)$

g)  $\cdot NOT \cdot F \cdot OR \cdot T \cdot AND \cdot 2 \cdot C \cdot LT \cdot -E+B$

h)  $T \cdot AND \cdot MOD(C, 2) \cdot EQ \cdot 0$

i)  $B \cdot GT \cdot A \cdot AND \cdot B-E \cdot NE \cdot I \cdot OR \cdot J \cdot K \cdot LT \cdot K+I \cdot OR \cdot \cdot NOT \cdot T$

Para verificar los resultados obtenidos, escribir un programa que asigne cada proposición a una variable lógica y luego escriba por pantalla el valor de la variable.

**Ejercicio 11.** Sean  $A$ ,  $B$  y  $C$  tres cadenas de caracteres, de longitud 4, 6 y 10 respectivamente, y  $\square$  un espacio en blanco. Determinar los valores de las siguientes asignaciones y operaciones:

a)  $A = 'PAR\square'$

b)  $B = 'PARE'$

c)  $A = 'PARED'$

d)  $B = '\square\square PAREDES'$

e)  $C = 'ASTRO' // '\square Y \square' // 'METEO'$

f)  $C = 'FERRO' // 'CARRILES'$

Una forma útil de encarar un problema, independientemente del lenguaje que usemos para construir nuestro programa, es escribir el algoritmo usando un pseudocódigo.

Un pseudocódigo es un lenguaje que nos permite escribir algoritmos en donde las instrucciones a seguir se dan de forma similar a como las diríamos con nuestras palabras.

Recordemos el programa fuente *Volumen.f*. Este programa en pseudocódigo se podría escribir de la siguiente forma.

- Leer el radio de la esfera.
- Tomar  $\pi = 3.141593$
- Calcular volumen  $= \frac{4}{3} \times \pi \times \text{radio}^3$
- Imprimir el volumen de la esfera.

De la misma forma para el programa fuente *Triangulo.f*, en pseudocódigo escribiríamos:

- Leer los lados del triángulo  $l_1$ ,  $l_2$  y  $l_3$ .
- Calcular el valor de  $s = \frac{1}{2} \times (l_1 + l_2 + l_3)$ .
- Calcular área  $= \sqrt{s \times (s - l_1) \times (s - l_2) \times (s - l_3)}$
- Imprimir el área del triángulo.

Antes de programar los siguientes ejercicios, escribir los correspondientes pseudocódigos.

**Ejercicio 12.** Construir un programa que permita calcular la velocidad  $\beta$ , el factor de Lorentz  $\gamma$  y el factor Doppler  $\delta$  dados por:

$$\beta = v/c, \quad \gamma = \frac{1}{\sqrt{1 - v^2/c^2}} \quad \text{y} \quad \delta = \frac{1}{\gamma(1 - \beta \cos(\theta))}.$$

Los datos para el cálculo son:  $v = 2.91 \times 10^5$  km/seg,  $\theta = 1.22^\circ$ . Puede utilizarse la instrucción *parameter* para asignarle valores a las constantes.

**Nota:** Recordar que las funciones trigonométricas intrínsecas usan radianes.

**Ejercicio 13.** Las ecuaciones de movimiento de una partícula en el espacio vienen dadas por:

$$\begin{cases} x = \alpha \cos^3(\theta) \\ y = \alpha \operatorname{sen}^3(\theta) \\ z = e^{\sqrt[4]{|x+y|}} \log(1 + \tan^2(\theta)) \end{cases}$$

donde  $0 \leq \theta \leq 2\pi$ . Realizar un programa que calcule a  $x$ ,  $y$  y  $z$ : a) si  $\alpha$  y  $\theta$  tienen sus valores asignados por programa; b) para pares de valores  $\alpha$  y  $\theta$  ingresados por teclado.

**Ejercicio 14.** Construir un programa para calcular las raíces  $x_1$  y  $x_2$  de un polinomio de grado dos,  $p(x) = ax^2 + bx + c$ , usando para su cálculo las siguientes expresiones, matemáticamente equivalentes:

Expresión 1:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Expresión 2:

$$x_1 = \frac{q}{a}, \quad x_2 = \frac{c}{q}, \quad q = -\frac{1}{2} \left( b + \operatorname{sgn}(b) \sqrt{b^2 - 4ac} \right)$$

$\operatorname{sgn}(\cdot)$  es la función signo. Usar los siguientes conjuntos de datos:

a)  $a = 1, \quad b = 1, \quad c = -2$

b)  $a = -1.2, \quad b = 5.3, \quad c = 4.8$

c)  $a = 3, \quad b = 4, \quad c = 5$

d)  $a = 10^{-3}, \quad b = 50, \quad c = 10^{-3}$

e)  $a = 10^{-5}, \quad b = 32, \quad c = 1$

**Nota:** En FORTRAN no existe la función *sgn* que aparece en la expresión de  $q$ . En FORTRAN tenemos una función de dos argumentos (*sign(arg1, arg2)*) que nos devuelve el valor absoluto de *arg1* con el signo que tiene *arg2*.