

INTRODUCCION A LA COMPUTACIÓN
PRÁCTICO – 4
PROGRAMACIÓN (FORTRAN) - INTRODUCCIÓN

VARIABLES, TIPOS DE DATOS INTRÍNSICOS Y EXPRESIONES

1- En un programa fortran llamado *varsAndTypes*:

- a) Declare dos variables de tipo entero *int1* , *int2*, con una sola sentencia.
- b) Declare dos variables enteras *int3* e *int4* e instanciadas con los valores 3 y 4 respectivamente.
- c) Declare dos variables de tipo real: *pi*, *e* y dos de tipo compleja: *complex1*, *complex2*. Instanciar *complex2* con parte real 0.5 y parte imaginaria 1.0.
- d) Declare luego tres variables de tipo lógico: *bool1*, *bool2*, *bool3*. Instanciarlas con los valores *true*, *false*, *false*, respectivamente.
- e) Declare dos variables de tipo caracter llamadas *char1* y *char2*. La variable *char2* debe ser de una longitud máxima de 15 caracteres, *char1* de un solo caracter.
- f) Instanciar (luego de la declaración de las variables):
 - *int1*, *int2* con los valores 15 y 6.
 - *pi*, *e*, con los valores 3.1416 y 2.71
 - *complex1* con (1,2)
 - *char1* con “S”, *char2* con “Ciudad de Montevideo”
- g) Imprima en pantalla un mensaje que diga “La variable <nombre de a variable definida> es: “, seguido por el valor de dicha variable. Haga esto para cada variable definida (solo una de cada tipo).

2 – En un programa llamado *expressions* (agregue las declaraciones necesarias en cada caso) :

- a) Realice la suma de dos variables enteras *int1*=3 e *int2* =8 y despliegue el resultado por pantalla.
- b) Elevar *int1* a la potencia *int2* de la parte a) y desplegar el resultado.
- c) Definir el real *real1*. Asignarle el resultado de la expresión aritmética: **$\pi \cdot (int1 + int2) / e$** con *pi* y *e* del la parte 1.
- d) Luego asignarle a *real1* el resultado de la siguiente expresión aritmética: **$1/3$** ¿Es el resultado esperado?. Asignarle a *real1* el resultado de la expresión aritmética **$1./3.$** , ¿el resultado es el esperado?

e) Expresiones lógicas.

Asigne a las variables lógicas *b1*, *b2*, *b3* los valores: **.true.**, **.false.**, **.false.** respectivamente. Asignarle a la variable lógica *b4* (e imprimir a continuación su valor e pantalla y analizar el resultado) las expresiones:

- $(b1.or.b2).and.b3$
- $(b1.and.b2).or.b1$
- $.not((b1.and.b2).or.b1)$
- $(b1.or.b2).and.(b1.or.b3)$
- $(.not.b1).or.(b1)$

f) Expresiones relacionales.

Asignarle a la variable lógica *b4* (e imprimir a continuación su valor e pantalla y analizar el resultado) las expresiones:

- $int1 < int2$
- $int1 + 5 \leq int2$
- $int1 == int2$
- $(int1 / 2.0) < 3$
- $(int1 / 2) == 1$

g) Expresiones compuestas (relacionales-lógicas).

Asignarle a la variable lógica *b4* (e imprimir a continuación su valor e pantalla y analizar el resultado) las expresiones:

- $((int1 + int2) \geq 10).and.false$
- $(int1 * int2) < (int1 + int2)$
- $(int1 < int2).or.(int > int2)$
- $(int1 < int2).and.(int > int2)$

CONTROL DE FLUJO

3 - a) Asignar las variables $int1=2$, $int2=3$, $int3=7$. Desplegar un mensaje en pantalla “ $int1 * int2$ es menor que $int3$ ” en caso de serlo o “ $int1 * int2 \geq int3$ ” si no es cierto.

b) Desplegar en pantalla “ $int1 + int3 = 5$ ” si es cierto, sino desplegar en pantalla “ $int1 + int2$ es distinto de 5”. En caso de ser cierto la primera condición, si $int1 * int2$ es mayor que $int3 - int2$, desplegar: “ $int1 + int2$ es igual a 5 y $int1 * int2$ es mayor que $int3 - int2$ ”.

c) Imprimir en pantalla la secuencia de números del 1 al 20, antecedida por el mensaje: “Número: ”.

d) Idem parte c) pero que la secuencia salte de a 3, esto es 1,4,9,etc.

e) Para la secuencia de números del 5 al 20, imprimir cada número multiplicado por 2 y sumado a 2.

- d) Definir una variable *maxLines* = 10. Luego imprimir en pantalla *maxLines* líneas, de forma que cada línea imprima el rango de enteros positivos que va de 1 a *maxLines*.
- g) Despliegue en pantalla “La expresiónA es mayor que la B” mientras que la expresión entera **exprA=20*i** sea mayor que la **exprB=3**i**, siendo i un entero positivo que comienza en 1. Imprimir el valor de i que hace que la *exprA* pase a ser menor que *exprB*.