

# Práctico 1 - Componentes básicos

## Programación 1 InCo - Facultad de Ingeniería, Udelar

### 1. Identificadores.

(a) Determine cuáles de los siguientes identificadores son válidos

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> <b>Promedio</b> | <input type="checkbox"/> \$XYZ                   |
| <input checked="" type="checkbox"/> <b>A1234</b>    | <input type="checkbox"/> A*b                     |
| <input type="checkbox"/> Program                    | <input checked="" type="checkbox"/> <b>Jorge</b> |
| <input type="checkbox"/> 1234A                      | <input type="checkbox"/> 506-74-3981             |
| <input type="checkbox"/> Can Can                    | <input type="checkbox"/> 4 x2                    |

(b) Explique por qué **no** sería correcto usar los siguientes identificadores para sus definiciones

I) 1986

Comienza con un dígito

Es el identificador de un tipo estándar (está permitido pero no es apropiado)

II) Vel\*Tiempo

Incluye un símbolo

VI) Var

Es una palabra reservada

III) End

Es una palabra reservada

VII) Sqrt

Es el identificador de una función estándar (está permitido pero no es apropiado)

IV) 1End

Comienza con un dígito

VIII) \$EFECTIVO

Comienza con un símbolo

v) Integer

### 2. Declaraciones de constantes y variables.

(a) Determine cuáles de las siguientes declaraciones de constantes son válidas:

- |  |
|--|
| <input checked="" type="checkbox"/> CONST impuesto = 0.09; |
| <input type="checkbox"/> CONST pi := 3.14;                 |
| <input checked="" type="checkbox"/> CONST diez = 9;        |
| <input type="checkbox"/> \$Acalif := 90;                   |

(b) Determine cuáles de las siguientes declaraciones de variables son válidas.

- VAR num1, num2 : Real;
- VAR num1; num2 : Integer;
- VAR total, suma, cuenta :  
Integer, Real;
- VAR idestudiante,  
numsegsocial : Integer;
- VAR num1, num2 : Real;
- var num1, num2:Integer;

3. Valores válidos de tipos básicos.

(a) Determine cuáles de los siguientes ejemplos son valores válidos de tipo **entero** en Pascal.

- 189**
- 2.5
- '33'
- 6,632
- 5555**
- 2.5e03
- 199.
- +199**
- maxint**

(b) Determine cuáles de los siguientes ejemplos son valores válidos de tipo **real** en Pascal.

- 0.01**
- .025
- 3.6**
- 3.6e-06**
- 3.e-06
- +8.3e2**
- 1.0e1.
- maxreal

(c) Determine cuáles de los siguientes ejemplos son valores válidos de tipo **carácter** en Pascal.

- A
- 'A'
- 'CAT'
- '8'
- '?'
- '??'

(d) Determine cuáles de los siguientes valores pueden aparecer en una declaración de constante. Determine el tipo de cada uno de ellos.

- 'e'  
 e
  - false
  - 999
  - 3.e
  - maxint
  - 0.000
  - '?'
  - !
- Char
- Boolean
- Integer
- Integer
- Real
- Char

4. Dada la siguiente declaración de variables:

VAR

```
temp, valor : Integer;  
num, suma  : Real;
```

Determine cuáles de las siguientes instrucciones de asignación son válidas.

num := temp + valor

num := valor / temp

valor := num + temp

valor := num / temp

num := suma

valor := num + 3

num := num DIV valor

5. Para cada una de las siguientes expresiones, determine su tipo y evalúela (calcule su valor).

i)  $6 \text{ DIV } 2 - 6 \text{ MOD } 5$

v)  $5 \text{ MOD } 8 + 8 \text{ MOD } 5$

Integer, 2

Integer, 8

ii)  $14 \text{ MOD } 2 * 6 + 3$

vi)  $7 \text{ MOD } 2 + 13 \text{ DIV } 3 - 2.5$

Integer, 3

Real, 2.5

iii)  $3 + 14 \text{ MOD } (2 * 3)$

vii)  $6 - 2 \text{ MOD } (1 + 4) + 5$

Integer, 5

Integer, 9

iv)  $3.2 + 14 \text{ MOD } (2 * 3)$

viii)  $6 - 2.1 * 3 + 1$

Real, 5.2

Real, 0.70

6. Dada la siguiente declaración de variables:

VAR

```
acosto, bcosto, ccosto, dcosto: real;
```

Considere las siguientes instrucciones de asignación que inicializan dichas variables:

```
acosto := 4.0;  
bcosto := 1.0;  
ccosto := -2.0;  
dcosto := 5.5
```

Evalúe cada una de las siguientes expresiones:

I) `sqrt (16)`

4.0

12

II) `trunc (-13.8)`

-13

VII) `trunc (10.1)`

10

III) `round (10.7)`

11

VIII) `trunc (8.6) - round (8.6)`

-1

IV) `sqr (5)`

25

IX) `sqrt (acosto / bcosto - ccosto + dcosto - 2.5)`

3.0

V) `round (-3.5)`

-4

X) `trunc (dcosto) * abs (ccosto * (bcosto / acosto))`

2.5

VI) `abs (-12)`

7. Dada la siguiente declaración de variables:

VAR

```
num, suma, total : Integer;  
valor             : Real;  
car1, car2       : Char;
```

Determine cuáles de las siguientes instrucciones de asignación son válidas.

`num := num + num`

`valor := total + car1`

`num := suma / total`

`car2 := car1`

`suma := num / total`

`car2 := car1 + 1`

`valor := total`

`valor := total * num + suma`

`car2 := 'car1'`

8. Para cada una de las siguientes expresiones, determine su tipo y evalúela (calcule su valor).

I) `5 + sqr(3) - 4 + trunc(3.6 - 2.1)`

Integer, 11

II) `3 * sqrt(8 MOD 6 * 10 DIV 5)`

Real, 6.0

III)  $6 + 9 * 8 \text{ DIV } 2 * \text{round}(1.362) - 2 * 3$

Integer, 36

IV)  $\text{trunc}(12 / 5 * \text{sqrt}(4 + 4 * 3) / 4)$

Integer, 2

9. ¿Cuál será la salida del siguiente programa al ejecutarlo en Pascal estándar? ¿y en Free Pascal?

```
PROGRAM Verificar;
VAR
  i : Integer;
  r : Real;
BEGIN
  r := i + 1;
  writeln (i,r)
END.
```

En Pascal estándar no está definido ningún valor inicial para las variables. Por lo tanto no podemos predecir cuál va a ser el valor que tiene la variable *i* cuando se realiza la asignación. Por lo tanto, no sabemos cuál va a ser la salida.

En el compilador Free Pascal, la variable *i* es inicializada por defecto con el valor 0 entonces la ejecución del programa despliega en la salida estándar los valores 0 y 1.

En este curso, se trabaja con Pascal estándar (tal como lo describe el libro) y **se considera un error** no inicializar las variables que son utilizadas en un programa.

10. Para cada uno de los siguientes programas en Pascal, trate de anticipar cuál va a ser la salida que emitirá al ejecutarlo. Después, ejecútelo en máquina y compare lo que esperaba con la salida que se exhibe. Ejecute el segundo programa varias veces, y use números tanto positivos como negativos, con partes fraccionarias mayores, menores, y exactamente iguales que 0.5. Cuando el programa espere el dato de entrada, introduzca su número real y digite ENTER.

```
PROGRAM Muestra1;
CONST
  pi = 3.1415926535;
  r1 = 2.0;
  r2 = 5.0;
VAR
  area: Real;
BEGIN
```

```

    area := pi * r1 * r1;
    writeln (r1, area);
    area := pi * sqr (r2);
    writeln (r2, area)
END.

```

Compile y ejecute el programa para obtener la solución.

```

PROGRAM Muestra2;
CONST
    medio = 0.5;
VAR
    num  : Real;
    a, b : Integer;
BEGIN
    readLn (num);
    a := round (num);
    b := trunc (num + medio);
    writeln (num, a, b)
END.

```

Compile y ejecute el programa para obtener la solución.

11. Considere las siguientes dos propiedades:

I)  $\ln(a^b) = b \times \ln(a)$

II)  $e^{\ln(x)} = x$

donde  $a$  y  $b$  son reales y  $a > 0$ .

Utilizando dichas propiedades, el operador de multiplicación (\*) y las funciones estándar de Pascal *ln* y *exp* escriba una expresión en Pascal que produzca el valor de  $a^b$ . Luego escriba un programa que evalúe dicha expresión y emita el resultado en la salida estándar. Los valores para  $a$  y  $b$  serán leídos desde la entrada estándar.

La descripción de las funciones *exp* y *ln* se encuentra en el **Apéndice D** del libro del curso, bajo el título **Funciones y Procedimientos Estándar (pag. 584)**.

```

PROGRAM Pr1Ej11;

VAR
    a,b: Integer;
    res: Real;
BEGIN
    Writeln('Ingresar dos valores enteros a y b');

```

```

Readln(a,b);

(* Calcular a^b en función de *, ln y exp:
   Dado que  $\ln(a^b) = b * \ln(a)$  y
   que  $e^{\ln(x)} = x$ , para todo x,
   entonces  $a^b = e^{(b * \ln(a))}$ 
*)
res := exp(b * ln(a));

Writeln('El valor de a^b es: ',res)
END.

```

12. En este ejercicio suponga que los datos de entrada serán siempre válidos, los programas a implementar no necesitan efectuar ese control. Verifique que los programas definidos funcionen correctamente para diferentes datos de entrada. Aplique todas las herramientas conceptuales y de Pascal que ha aprendido hasta el momento.

i) Escriba un programa que calcule el período de un péndulo de longitud  $l$  y emita el resultado en la salida estándar. El período de un péndulo está dado por la siguiente ecuación  $t = 2 \times \pi \times \sqrt{l/g}$  donde  $\pi = 3,14$ ,  $g = 9,8$  y  $l$  es leído de la entrada estándar.

```

PROGRAM Pr1Ej12a;
CONST pi = 3.14;
      g = 9.8;
VAR
  l, per: Real;
BEGIN
  Writeln('Introduzca la longitud del pendulo: ');
  Readln(l);

  (* Se calcula el periodo para la longitud ingresada *)
  per := 2 * pi * sqrt(l/g);

  Writeln('El periodo es: ',per)
END.

```

ii) Escriba programa que calcule el área de un triángulo cuyos lados son de longitud  $a$ ,  $b$  y  $c$  (los cuales serán leídos de la entrada estándar) y emita el resultado en la salida estándar. El área  $A$  de un triángulo se puede calcular a partir de sus lados usando la fórmula  $A = \sqrt{s \times (s - a) \times (s - b) \times (s - c)}$  donde  $s = (a + b + c)/2$ .

```

PROGRAM Pr1Ej12b;
VAR
  lado1,lado2,lado3 : real;
  s, area : real;

```

```
BEGIN
```

```
Writeln('Inserte los lados del triángulo: ');
```

```
Readln(lado1,lado2,lado3);
```

```
(* Se calcula el área del triángulo según la fórmula *)
```

```
s := (lado1+lado2+lado3)/2;
```

```
area := sqrt(s * (s-lado1) * (s-lado2) * (s-lado3));
```

```
Writeln('El area es: ', area)
```

```
END.
```