

# Práctico 1 - Componentes básicos

## Programación 1 InCo - Facultad de Ingeniería, Udelar

### 1. Identificadores.

(a) Determine cuáles de los siguientes identificadores son válidos

- |                                   |                                      |
|-----------------------------------|--------------------------------------|
| <input type="checkbox"/> Promedio | <input type="checkbox"/> \$XYZ       |
| <input type="checkbox"/> A1234    | <input type="checkbox"/> A*b         |
| <input type="checkbox"/> Program  | <input type="checkbox"/> Jorge       |
| <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 <small>Sintaxis incorrecta - Debe comenzar con un Num</small>  | V) Integer <small>Palabra Reservada</small>                                     |
| II) Vel*Tiempo <small>* no es un carácter valido</small>               | VI) Var <small>Palabra Reservada</small>  |
| III) End <small>Palabra Reservada</small>                              | VII) Sqrt <small>Palabra Reservada</small>                                      |
| IV) 1End <small>Sintaxis incorrecta - Debe comenzar con un Num</small> | VIII) \$EFFECTIVO <small>Sintaxis incorrecta - Debe comenzar con un Num</small> |

### 2. Declaraciones de constantes y variables.

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

- ☐ `CONST impuesto = 0.09;`
- ☐ `CONST pi := 3.14;` el operado := se usa para asignar el valor a una variable luego de estar declarada
- ☐ `CONST diez = 9;`
- ☐ `$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;`
- ☐ `VARnum1, 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.

- |                                |                                 |
|--------------------------------|---------------------------------|
| <input type="checkbox"/> 189   | <input type="checkbox"/> 2.5e03 |
| <input type="checkbox"/> -2.5  | <input type="checkbox"/> 199.   |
| <input type="checkbox"/> '33'  | <input type="checkbox"/> +199   |
| <input type="checkbox"/> 6,632 | <input type="checkbox"/> maxint |
| <input type="checkbox"/> -5555 |                                 |

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

- |  |   |
|--|---|
| <input type="checkbox"/> -0.01   | <input type="checkbox"/> 3.e-06 <small>funciona pero no es valido, porque no hay dígito luego del punto</small> |
| <input type="checkbox"/> .025 <small>funciona pero no es valido porque no hay dígito antes del punto</small> | <input type="checkbox"/> +8.3e2   |
| <input type="checkbox"/> -3.6  | <input type="checkbox"/> 1.0e1. <small>el exponente no es Entero</small>  |
| <input type="checkbox"/> 3.6e-06   | <input type="checkbox"/> maxreal <small>no existe una constante llamada así</small>                             |

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

- |                                |                               |
|--------------------------------|-------------------------------|
| <input type="checkbox"/> A     | <input type="checkbox"/> '8'  |
| <input type="checkbox"/> 'A'   | <input type="checkbox"/> '?'  |
| <input type="checkbox"/> 'CAT' | <input type="checkbox"/> '??' |

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

- |                                |                                  |
|--------------------------------|----------------------------------|
| <input type="checkbox"/> 'e'   | <input type="checkbox"/> -maxint |
| <input type="checkbox"/> e     | <input type="checkbox"/> -0.000  |
| <input type="checkbox"/> false | <input type="checkbox"/> '?'     |
| <input type="checkbox"/> 999   | <input type="checkbox"/> !       |
| <input type="checkbox"/> 3.e   |                                  |

4. Dada la siguiente declaración de variables:

VAR

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

Determine cuáles de las siguientes instrucciones de asignación son válidas.

- |  |  |
|--|--|
| <input type="checkbox"/> num := temp + valor | <input type="checkbox"/> num := valor / temp <small>div real</small> |
| <input type="checkbox"/> valor := num + temp | <input type="checkbox"/> valor := num / temp                         |
| <input type="checkbox"/> num := suma         | <input type="checkbox"/> num := num DIV valor                        |
| <input type="checkbox"/> valor := num + 3    |  |

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

- |   |  |
|---|--|
| I) 6 DIV 2 - 6 MOD 5 <small>Integer, valor = 2</small>  | V) 5 MOD 8 + 8 MOD 5 <small>integer, valor = 8</small>         |
| II) 14 MOD 2 * 6 + 3 <small>Integer, valor 3</small>    | VI) 7 MOD 2 + 13 DIV 3 - 2.5 <small>real, valor = 2.5</small>  |
| III) 3 + 14 MOD (2 * 3) <small>Integer, valor 5</small> | VII) 6 - 2 MOD (1 + 4) + 5 <small>integer, valor = 9</small>   |
| IV) 3.2 + 14 MOD (2 * 3) <small>real, valor 5.2</small> | VIII) 6 - 2.1 * 3 + 1 <small>real, valor = 6.99x10(-1)</small> |

6. Dada la siguiente declaración de variables:

```
VAR
    acostó, bcosto, ccosto, dcosto: real;
```

Considere las siguientes instrucciones de asignación que inicializan dichas variables:

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

Evalúe cada una de las siguientes expresiones:

- |  |   |
|--|---|
| I) <code>sqrt (16)</code> valor = 4        | VII) <code>trunc (10.1)</code> valor = 10   |
| II) <code>trunc (-13.8)</code> valor = -13 | VIII) <code>trunc (8.6) - round (8.6)</code> valor = -1                           |
| III) <code>round (10.7)</code> valor = 11  | IX) <code>sqrt (acostó / bcosto - ccosto + dcosto - 2.5)</code> el valor es = 3   |
| IV) <code>sqr (5)</code> valor = 25        | X) <code>trunc (dcosto) * abs (ccosto * (bcosto / acostó))</code> el valor es = 3 |
| V) <code>round (-3.5)</code> valor = -4    |   |
| VI) <code>abs (-12)</code> valor = 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.

- |  |  |
|--|--|
| <input type="checkbox"/> <code>num := num + num</code> válida            | <input type="checkbox"/> <code>valor := total + car1</code> inválida |
| <input type="checkbox"/> <code>num := suma / total</code> inválida       | <input type="checkbox"/> <code>car2 := car1</code> válida            |
| <input type="checkbox"/> <code>suma := num / total</code> inválida       | <input type="checkbox"/> <code>car2 := car1 + 1</code> inválida      |
| <input type="checkbox"/> <code>valor := total</code> válida              | <input type="checkbox"/> <code>car2 := 'car1'</code>                 |
| <input type="checkbox"/> <code>valor := total * num + suma</code> válida |  |

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)` interge, al aplicar trunc este nos devuelve un numero entero valor es de 11
- II) `3 * sqrt(8 MOD 6 * 10 DIV 5)` real, debido a que es lo que devuelve la raíz . Valor 6
- III) `6 + 9 * 8 DIV 2 * round(1.362) - 2 * 3` integer, Valor = 36
- IV) `trunc(12 / 5 * sqrt(4 + 4 * 3) / 4)` interger, valor = 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.
```

El programa lograra compilar y dará aviso de que existe una variable sin inicializar antes de ser utilizada 'i'

valor = (0 , 1)

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.
```

```
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.
```

11. Considere las siguientes dos propiedades:

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

$\ln(a^b) = b * \ln(a)$

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

$e^{\ln(a^b)} = e^{(b * \ln(a))} \rightarrow \text{elevo a la e}$

$a^b = e^{(b * \ln(a))}$

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)**.

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.
- 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  $s = (a + b + c)/2$ .

```
{
  Fórmula de Herón
  Á = sqrt(numS * (numS - numA) * (numS - numB) * (numS - numC));

  Condición
  La suma de cualquier par de lados de un triángulo debe ser mayor que el tercer lado.
}
program Triangulo;
var
  numA, numB, numC, numS, area : real;
begin
  writeln('Ingrese a continuacion el valor de a, b y c:');
  readln(numA);
  readln(numB);
  readln(numC);

  writeln('a = ', numA:0:2);
  writeln('b = ', numB:0:2);
  writeln('c = ', numC:0:2);

  (* Se verifica que se cumpla la condicion para un triangulo valido*)
  if (numA + numB > numC) and (numA + numC > numB) and (numB + numC > numA) then
    begin
      numS := (numA + numB + numC) / 2;
      area := sqrt(numS * (numS - numA) * (numS - numB) * (numS - numC));
      writeln('El area del triangulo es:', area:0:2);
    end
  else
    begin
      writeln('Error: Los lados ingresados no forman un triangulo valido. ');
      writeln('La suma de cualquier par de lados de un triángulo debe ser mayor que el tercer lado. ');
    end;
end.
```