

Solución orientativa Tarea 1.- Módulo: PROGRAMACIÓN. Curso 2017/18.

Ejercicio 1

Para cada una de las siguientes instrucciones, escribe una expresión equivalente. Por ejemplo, dada una expresión $A = A + 1$, una expresión equivalente podría ser: $A++$.

APARTADO	EXPRESIÓN	EXPRESIÓN EQUIVALENTE
a)	$H = H + 1 - 3$	$H = H - 2$
b)	$V /= T$	$V = V / T$
c)	$A = 4 / B + 7 * C / 5 - 1 * 2$	$A = 4 / B + 7 * C / 5 - 2$
d)	$D = D * 5 - 8 / 4 - 2$	$D = D * 5 - 2 - 2$

Ejercicio 2

Indica si las expresiones siguientes son correctas, teniendo en cuenta que las variables **a** y **b** son de tipo entero (int), **c** es de tipo real (double), **d** de tipo lógico (boolean), **e** de tipo carácter (char) y **f** de tipo cadena de caracteres (string). En todo caso debes justificar la respuesta:

Apartado	Expresión	Respuesta
a)	"pincha y" + "corta"	Correcto, se concatenan las cadenas de caracteres y se genera un nuevo String.
b)	$f = \text{true}$	No es correcto, debería ser "true" o bien que f fuera de tipo boolean
c)	$a = 2.5$	No es correcto, a debería ser de tipo double, pero es de tipo int. En este caso, no se hace una conversión automática de tipos, dado que hay una pérdida de precisión de int a double.
d)	$d = \text{false}$	Correcto. La variable "d" es de tipo boolean y se le puede asignar false.
e)	$b = 25 \% 4$	Correcto. La variable "b" es entera y se le puede asignar un entero (el resultado de la

		operación módulo es un número entero).
f)	<code>e = "rony"</code>	No es correcto, un String no se puede convertir en char.
g)	<code>a = 5</code>	Correcto. La variable "a" es de tipo entero y se le puede asignar un entero.
h)	<code>a = "b" + "d" + "t"</code>	Incorrecto, es una "suma" de Strings, por lo que el resultado es la concatenación de los tres Strings
i)	<code>b = 1.0 + 2</code>	Incorrecto, produce un error de incompatibilidad de tipos. <code>1.0 + 2</code> se convierte a double - 3.0 - y no se puede asignar a b, dado que esta es de tipo entero y se produce una pérdida de precisión. Lo correcto sería: <code>b=(int)(1.0 + 2);</code>
j)	<code>c = ++b</code>	Correcto, dado que c, al ser de un tipo de mayor precisión (double), puede recoger un valor de un tipo de menor precisión (b es int).

Ejercicio 3

Evalúa las siguientes expresiones lógicas y/o numéricas teniendo en cuenta la precedencia de los operadores e indica el resultado. Por ejemplo, dada la expresión: `7 + 6 / 2 * 3 + 4 / 2 * 2`, el tipo sería Numérica, la evaluación sería: `7+9+4` y el resultado: 20.

Apartado	Expresión	Tipo	Evaluación	Resultado
a)	<code>8 >= 9 8 < 2 + 6</code>	Lógica	<code>false false</code>	false
b)	<code>2 * 7 + 12 / 3 / 4 + 3 * 2</code>	Numérica	<code>14 + 1 + 6</code>	21
c)	<code>8 != 9 - 1 && 6 * 4 >= 12</code>	Lógica	<code>false && true</code>	false
d)	<code>d = 1000 / 10 + 40 * 10</code>	Numérica	<code>(100 + 400)</code>	500

Ejercicio 4.

Usa **pseudocódigo** para dar solución al siguiente problema:

Pedir cuatro valores reales leyéndolos por teclado, y escribirlos por pantalla en el orden inverso al que se han leído.

Algoritmo EscribirInversamente

Var num1, num2, num3, num4 : real

Inicio

 Escribir "Escriba el primer número (de 4 en total)"
 Leer num1

 Escribir "Escriba el segundo número (de 4 en total)"
 Leer num2

 Escribir "Escriba el tercer número (de 4 en total)"
 Leer num3

 Escribir "Escriba el cuarto número (de 4 en total)"
 Leer num4

 Escribir ("El último número leído es: ", num4)
 Escribir ("El penúltimo número leído es: ", num3)
 Escribir ("El antepenúltimo número leído es: ", num2)
 Escribir ("El primer número leído es: ", num1)

Fin

Ejercicio 5.

Utiliza pseudocódigo para solucionar el siguiente problema:

Pedir al usuario el valor de x y calcular el valor de la ecuación $y = 7 \cdot x^3 + 87$. Escribir el valor de y .

Algoritmo ResolverEcuacion

Var x, y : real

Inicio

 Escribir "Escriba el valor de x"
 Leer x

$y \leftarrow (7 * (x * x * x)) + 87$

 Escribir ("El resultado es: ", y)

Fin

Ejercicio 6.

Escribe el algoritmo en pseudocódigo para dar solución al siguiente problema:

Se pedirá al usuario un número entero por teclado. El programa debe averiguar si el número cumple con las dos condiciones siguientes: si está comprendido entre 275 y 380. Si lo está, entonces comprobará si el número es divisible por 3. Si el número No está comprendido entre 275 y 380, entonces lo que comprobará será si es divisible por 4. Por tanto, por ejemplo si el número leído fuera el 300, el algoritmo escribiría "El número está comprendido en el intervalo." y a continuación escribiría "El número es divisible por 3."

Los dos siguientes casos se consideran correctos:

Algoritmo Entre275y380

Var numero : entero

Inicio

Escribir "Escriba el número a comprobar"

Leer numero

SI (numero >= 275 y numero <= 380) ENTONCES

Escribir "El número está comprendido en el intervalo."

SI (numero % 3 = 0)

Escribir "El número es divisible por 3."

FIN_SI

SINO

Escribir "El número NO está en el intervalo."

SI (numero % 4 = 0)

Escribir "El número es divisible por 4."

FIN_SI

FIN_SI

Fin

Algoritmo Entre275y380

Var numero : entero

Inicio

Escribir "Escriba el número a comprobar"

Leer numero

SI (numero >= 275 y numero <= 380) ENTONCES

Escribir "El número está comprendido en el intervalo."

SI (numero % 3 = 0)

Escribir "El número es divisible por 3."

```
SINO
    Escribir "El número no es divisible por 3."
FIN_SI
SINO
    Escribir "El número NO está en el intervalo."
    SI (numero % 4 = 0)
        Escribir "El número es divisible por 4."
        SINO
            Escribir "El número no es divisible por 4."
        FIN_SI
    FIN_SI
FIN_SI
```

Fin

Ejercicio 7.

Haz un algoritmo en pseudocódigo que lea dos números por teclado. Si el primer número es menor o igual que el segundo, entonces se escribirán por pantalla los números desde el primero al segundo.

```
Algoritmo EscribirEntreIntervalos
/* Declaración de variables. De tipo entero ya que no necesitamos decimales.*/
Var
    contador : entero
    numero: entero
    numero2: entero

Inicio
    Escribir "Escriba el primer número"
    Leer numero

    Escribir "Escriba el segundo número"
    Leer numero2

    /* Si el primer número es menor o igual que el segundo */
    SI (numero <= numero2) ENTONCES
        /* Escribir los números comprendidos entre los números leídos */
        PARA contador <- numero HASTA numero2 HACER
            /* Escribimos el resultado */
            Escribir (contador)
        FIN_PARA
    FIN_SI
```

Fin



Otra solución posible podría ser:

Algoritmo EscribirEntreIntervalos

Var

 numero: entero
 numero2: entero

Inicio

 Escribir "Escriba el primer número"
 Leer numero1

 Escribir "Escriba el segundo número"
 Leer numero2

 Mientras numero1<=numero2 Hacer
 mostrar numero1
 numero1<-numero1+1
 FinMientras

FinAlgoritmo

Ejercicio 8.

Haz un algoritmo en **pseudocódigo**, que esté iterando continuamente, pidiendo valores enteros por teclado mientras el valor leído sea mayor que 0. En el momento en que sea 0 o menor que 0, el programa finalizará la ejecución. Obviamente se debe usar un bucle para ese cometido.

Según el número que se lea en cada iteración, el programa informará sobre el tipo de ángulo: agudo si el valor está entre 0 y menos de 90, recto si es igual a 90 y obtuso si es mayor de 90.

```
Algoritmo Angulos
  Var
    angulo : entero

Inicio
  angulo <- 0

  MIENTRAS (angulo >= 0) HACER

    Escribir "Escriba un valor de ángulo: "
    Leer angulo

    SI (angulo >= 0 y angulo < 90) ENTONCES
      Escribir("Es un ángulo agudo.")
    FIN_SI
    SI (angulo = 90) ENTONCES
      Escribir("Es un ángulo recto.")
    FIN_SI
    SI (angulo > 90) ENTONCES
      Escribir("Es un ángulo obtuso.")
    FIN_SI

  FIN_MIENTRAS

  Escribir "Fin (el ángulo introducido es menor de cero)"

Fin
```

Ejercicio 9.

Realiza una pequeña aplicación en **java** (utilizando NetBeans) que dé solución al siguiente problema:

Resuelto en proyecto adjunto.