

## EJERCICIOS DE REPASO – PSEUDOCÓDIGO Y JAVA BÁSICO

### 2.1 Saludo

Realiza un programa que solicita un nombre por teclado y luego muestra por pantalla el texto "Hola " seguido del nombre introducido.

- Ejemplo de ejecución:

```
Introduzca su nombre: Pedro
Hola Pedro
```

En el ejemplo anterior el texto en color verde ha sido escrito por el usuario en verde, mientras que el resto lo ha escrito el propio programa.

### 2.2 Suma

Realiza un programa que imprime el texto "2 + 2 = " con una instrucción y con otra instrucción imprime el resultado de la operación aritmética 2 + 2

- Resultado de la ejecución:

```
2 + 2 = 4
```

### 2.3 Intercambio

Realiza un programa que declare dos variables num1 y num2 de tipo entero, se impriman mostrando su nombre y valor, luego se deben intercambiar los valores de dichas variables para finalmente, volver a mostrarse.

- Ejemplo de ejecución:

```
num1=5 y num2=8
num1=8 y num2=5
```

Nota: La instrucción para imprimir los valores en la primera línea debe tener el mismo código que para la segunda línea

### 2.4 Multiplica constante

Realiza un programa que declare una variable de tipo double con valor 6.5 y una constante del mismo tipo y valor 7.8, mostrando por consola su multiplicación con dos decimales utilizando el operador aritmético \*. Explica el resultado

- Resultado de la ejecución:

```
50,70
```

## 2.5 División con decimales

Realiza un programa que declare dos variables n1 y n2 de tipo entero con los valores 5 y 2 respectivamente y muestre su división incluyendo los decimales

➤ Ejemplo de ejecución:

```
n1=5
n2=2
n1/n2= 2.5
```

## 2.6 Operadores aritméticos

Realizar un programa que declare dos variables de tipo byte con los valores que quieras y muestre su suma, resta, multiplicación y división, utilizando operadores aritméticos. Se debe mostrar por pantalla cada operación en una línea distinta. Nota: La división no hace falta que tenga decimales.

➤ Ejemplo de ejecución:

```
4 + 2 = 6
4 - 2 = 2
4 * 2 = 8
4 / 1 = 2
```

## 2.7 División y módulo

Realizar un programa que muestre por consola el resultado de las expresiones 45/10 y 45%10. Con comentarios en el código, explica lo que muestra y trata de buscarles una utilidad en el tratamiento de números.

➤ Ejemplo de ejecución:

```
45 / 10 = 4
45 % 10 = 5
```

## 2.8 Tabla de multiplicar

Realiza un programa que solicite por teclado un número [1-10] y muestre por pantalla la tabla de multiplicar de dicho número.

➤ Ejemplo de ejecución:

```
Introduzca un número [1-10]: 4
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
```

## 2.9 Edad y peso

Realiza un programa que solicite por teclado la edad (entero) y el peso (con decimales) y muestre la edad con un año más y el doble del peso, mostrando un mensaje con los resultados en una única línea.

- Ejemplo de ejecución:

```
Indica tu edad: 23
Indica tu peso: 67,2
Con un año más tendrás 24 años y el doble de tu peso sería 134.4
kg
```

## 2.10 Hora del día (if)

Utilizando la instrucción **if**, realiza un programa que pida una hora por teclado y que muestre luego “Buenos días”, “Buenas tardes” o “Buenas noches” según la hora.

Se utilizarán los tramos de 6 a 12, de 13 a 20 y de 21 a 5 respectivamente. Si el número no está en el rango (0-23) se indicará que no es correcto.

Sólo se tienen en cuenta las horas, los minutos no se deben introducir por teclado.

- Ejemplo de ejecución:

```
Por favor, introduzca una hora del día (0 - 23): 4
Buenas noches
```

Nota: En el ejemplo anterior el texto en color verde ha sido escrito por el usuario en verde, mientras que el resto lo ha escrito el propio programa.

## 2.11 Día de la semana (switch)

Utilizando la instrucción **switch**, escribe un programa en que dado un número del 1 a 7 escriba el correspondiente nombre del día de la semana. En caso de introducir un número fuera del rango, se informará al usuario de su error.

- Resultado de la ejecución:

```
Por favor, introduzca un número del 1 al 7: 6
Sábado
```

## 2.12 Cantidad de 5 (while)

Utilizando la instrucción **while**, realiza un programa que solicite un número entre 1 y 1.000.000 y muestre la cantidad de dígitos 5 que contiene.

- Ejemplo de ejecución:

```
Introduce un número (1-1.000.000): 356574
2
```

### 2.13 Hay 10 (do-while)

Utilizando la instrucción **do-while**, realiza un programa que lea una secuencia de notas (con valores que van de 0 a 10) que termina con el valor -1 e imprima "SI" si ha habido algún 10 o "NO" en caso contrario.

➤ Resultado de la ejecución:

```
Introduce un número (0-10): 3
Introduce un número (0-10): 1
Introduce un número (0-10): 10
Introduce un número (0-10): 2
Introduce un número (0-10): -1
SI
```

### 2.14 Mayor de 5 (for)

Utilizando la instrucción **for**, realiza un programa que lea por teclado 5 números enteros y determine cuál es el mayor.

➤ Ejemplo de ejecución:

```
Introduce el numero 1: 3
Introduce el numero 2: 8
Introduce el numero 3: 5
Introduce el numero 4: 7
Introduce el numero 5: 2
El mayor es: 8
```

### 2.15 En secuencia

Realiza un programa que solicite la cantidad de números a introducir (más de uno) y luego los pida por teclado (los números serán enteros). El programa debe determinar si los números están en secuencia ascendente, descendente o no hay secuencia. Están en secuencia ascendente cuando cada número es superior al anterior. Están en secuencia descendente cuando cada número es inferior al anterior. No están en secuencia cuando no se cumple ninguna de las dos posibilidades anteriores.

➤ Resultado de la ejecución 1:

```
Introduce la cantidad de numeros: 4
1
2
3
4
Secuencia ascendente
```

➤ Resultado de la ejecución 2:

```
Introduce la cantidad de números: 3
45
23
12
```

Secuencia descendente

- Resultado de la ejecución 3:

Introduce la cantidad de números: 3

4

-1

7

No hay secuencia

## 2.16 Préstamo

Realizar un programa que pida la cantidad del préstamo que solicitamos a una entidad bancaria. A partir de ese momento, se irán introduciendo valores que serán las distintas aportaciones que se hacen para devolver el préstamo. Por cada aportación el programa mostrará el importe pendiente. El programa termina cuando suma de las aportaciones sea igual o superior al préstamo solicitado. Todos los valores pueden ser número reales (con decimales)

- Ejemplo de ejecución 1:

Importe del préstamo: 2000

500

1500.0

1000

500.0

500

- Ejemplo de ejecución 2:

Importe del préstamo: 1000

200

800.0

345

455.0

500

## 2.17 El ascensor

Realiza un programa que calcule la cantidad de pisos recorridos por un ascensor. El trayecto siempre empieza y termina en el piso 0 o Bajo. El ascenso va desde el piso 0 al piso 20.

- Ejemplo de ejecución:

Introduzca los pisos (uno por línea)

4

2

6

3

0

16

### 2.18 Transportando agua

Realiza un programa que solicite una cantidad de litros de agua (número entero entre 1 y 1000) y las reparta entre botellas de 5, 2 y medio litro. Para facilitar el transporte, se debe tratar de ocupar el menor número posible de botellas. El programa debe indicar cuántas botellas de cada tipo se han utilizado. No se pueden utilizar bucles. Revisa la utilidad de los operadores % y /.

- Ejemplo de ejecución 1:

```
Cantidad de litros de agua (1-1000): 456
91 botellas de 5, 0 botellas de 2 y 2 botellas de 1/2
```

- Ejemplo de ejecución 2:

```
Cantidad de litros de agua (1-1000): 123
24 botellas de 5, 1 botellas de 2 y 2 botellas de 1/2
```

- Ejemplo de ejecución 3:

```
Cantidad de litros de agua (1-1000): 200
40 botellas de 5, 0 botellas de 2 y 0 botellas de 1/2
```

### 2.19 Número de cifras

Escribe un algoritmo en pseudocódigo y el código en Java que pida un número entero positivo por teclado (debéis controlar que es positivo) y muestre el número de cifras que tiene. Por ejemplo: si escribimos el número 34567, nos tiene que responder el programa diciendo que tiene 5 cifras.

### 2.20 Temperatura

Escribe un algoritmo en pseudocódigo y realizar el código en Java que lea una temperatura en grados Celsius y los pase a Fahrenheit. La fórmula para la conversión es:  $F = 32 + (9 * C / 5)$

### 2.21 Capicúa

Escribe un algoritmo en pseudocódigo y el código en Java que compruebe si un número de 3 cifras introducido por teclado es capicúa.

### 2.22 Capicúa

Escribe un algoritmo en pseudocódigo y el código en Java que lea por teclado tres números enteros H, M y S (horas, minutos y segundos) y comprobar que es una hora válida. Por ejemplo:

- Ejemplo de ejecución 1:

```
Introduce el valor de H: 10
Introduce el valor de M: 25
Introduce el valor de S: 34
Es una hora válida: Son las 10:25:34 horas
```

### 2.23 Amigos

Escribe un algoritmo en pseudocódigo y el código en Java que calcule si dos números son amigos. Dos números A y B son amigos si la suma de los divisores de A es igual a B y la suma de los divisores de B es igual a A. Por ejemplo 220 y 284.

Los divisores de 220 son: 1,2,4,5,10,11,20,22,44,55 y 110. La suma de estos divisores es 284.

Los divisores de 284 son: 1,2,4,71 y 142. La suma de estos divisores es 220.

Por lo tanto, los números 220 y 284 son amigos.

### 2.23 Perfecto

Escribe un algoritmo en pseudocódigo y el código en Java que calcule si un número es perfecto. Un número es perfecto si es igual a la suma de todos sus divisores positivos sin incluir el propio número. Por ejemplo, el número 6 es perfecto. Sus divisores son: 1,2,3 y 6. Al sumar todos menos el 6:  $1+2+3=6$

### 2.24 Narcisista

Escribe un algoritmo en pseudocódigo y el código en Java que calcule si un número es narcisista. Un número es narcisista si la suma de sus cifras elevada al número de cifras que tiene el número es igual al propio número.

Ejemplo:

153 es un número narcisista. El número tiene 3 cifras. Si sumamos cada cifra elevada a 3 nos da el propio número 153:

$$1^3+5^3+3^3=1+125+27= 153$$