

# Fundamentos de Programación

## Cuaderno de Trabajo 3

Grado en Sistemas de la Información, turno de mañana

Grado en Ingeniería Informática, turno de mañana

### Ejercicios resueltos

1. Escribe un programa capaz de identificar si un número leído por teclado es múltiplo de 3.

**SOLUCION PROPUESTA 1 (no modularizada):**

```
n = int(input('Introduce un numero: '))
if (n % 3 == 0):
    print("El numero introducido es multiplo de 3")
else:
    print("El numero introducido NO es multiplo de 3")
```

**SOLUCION PROPUESTA 2 (modularizada):**

```
def es_multiplo_de_3(n):
    """ int -> bool
    OBJ: Indica si el entero n es multiplo de 3 """
    return (n % 3 == 0)

#Probador
numero = int(input('Introduce un numero: '))
if es_multiplo_de_3(numero):
    print("El numero introducido es multiplo de 3")
else:
    print("El numero introducido NO es multiplo de 3")
```

2. Escribe una función y su programa principal para calcular el factorial de un número validado en el rango de 0 a 30. Cuando se pase el número n a la función factorial dicho número ya habrá sido validado.

**SOLUCION PROPUESTA:**

```
def factorial(n):
    """ int -> long
    OBJ: Calcula el factorial de un numero entero
    PRE: 0<=n<=30 """
    acu = 1
    for i in range(1,n+1):
        acu *= i
    return acu

#Probador
n = int(input("Introduce un número entero entre 0 y 30: "))
```

```

if 0<n<30:
    print(factorial(n))
else:
    print("Error, el número introducido no cumple las condiciones")

```

## Ejercicios propuestos

1. Escribe un código que implemente el siguiente comportamiento: "Si la compra es superior a 100EUR se aplica un descuento del 5% si se paga al contado, pero si el pago es con tarjeta sólo se aplica el 2%". Asegúrate de que el importe de la compra es un número válido antes de proceder a los cálculos (pista: usa try para comprobar que es posible convertir la entrada a un float).
2. Escribe un subprograma que calcule el producto de dos números enteros leídos desde el teclado sin utilizar el operador de multiplicación \*. Ej: si recibe 2 y 3 devuelve 6; si recibe 2 y 0 devuelve 0; si recibe -3 y 2 dará -6. El programa no asumirá que los números son mayores que cero, debiendo validar las entradas.
3. Escribe un programa que lea la hora en notación de 24 horas y la devuelva en notación de 12 horas (ejemplo: las 18:30 serán las 6:30 PM). Valida las entradas para asegurarte de que se trata de valores en el rango correcto.
4. Como habrás observado, muy a menudo necesitamos validar la información introducida por el usuario. Programa una función "validar\_entero" que se asegure de que una entrada del usuario es un entero. Ten en cuenta que dicha entrada puede ser cualquier cosa, por ejemplo un valor real, un booleano o incluso una cadena como "Hola". Nuestra función recibirá un texto y retornará verdadero si es un entero (dando por válida la entrada) o falso (rechazando la entrada).
5. Programa una función "validar\_real" similar a la del ejercicio anterior pero que valide que una entrada es un número real.
6. Escribe una función que a partir de un número real y de un número entero calcule el valor del primero elevado al segundo sin usar el operador potencia \*\*, ni la función pow(). Reutiliza las funciones de los apartados 5 y 6.
7. Una universidad acaba de modificar su sistema de representación de la calificación de los alumnos, que como es sabido son valores entre 0 y 10. A partir de ahora, se calificarán como "A" las notas mayores o iguales a 8,5, "B" las mayores o iguales a 6,5, "C" las calificaciones mayores o iguales a 5, "D" las calificaciones mayores o iguales a 3,5, y "F" todas las demás. Programa una función que reciba una calificación numérica y retorne la letra con la nueva calificación. Asegúrate de que la calificación introducida es válida (idea: programa una función lo suficientemente genérica que se pueda luego reutilizar en programas que necesiten una validación similar).
8. Escribe un algoritmo que tras leer tres enteros los escriba en pantalla en orden creciente. Como siempre, valida las entradas.
9. Ya has visto que la validación de entradas es algo tan presente que no podemos huir de ello, así que si te parece vamos a escribir una función que solicite al usuario introducir un entero y que no pare de pedirselo hasta que la información

introducida sea válida. La idea es usar la construcción `while` combinada con las funciones de validación programadas en el ejercicio 4, consiguiendo una función cuya cabecera sería la siguiente:

```
def leer_entero_validado():  
    """ nada --> int  
    OBJ: Solicita un entero al usuario, lo valida y lo retorna sólo  
    cuando se ha asegurado de que es realmente un entero  
    """
```

Ahora te toca a ti completarla, codificando sus instrucciones. ¡Adelante!

10. Codifica un subprograma que reciba un número entero, y si es entre 1 y 12 escriba un mensaje por pantalla indicando el mes a que dicho número corresponde. En caso contrario deberá mostrar un mensaje de error. Valida las entradas utilizando la función del ejercicio 9.
11. Programa una función que permita determinar si un determinado carácter es una vocal o no, utilizando la construcción "if-elif". Repite el ejercicio utilizando esta vez el operador booleano OR y observa las diferencias.
12. Escribe un programa que, después de preguntar cuántos números se van a introducir, pida esos números (enteros o reales) y devuelva su media aritmética, el mayor y el menor. El programa debe controlar que la cantidad de números es mayor de 2 y en caso contrario ha de mostrar un mensaje de error. Como siempre, valida las entradas.
13. Escribe un programa que lea una letra por teclado, correspondiente a un día de la semana (L,M,X,J,V,S,D), y muestre un menú de comida para dicho día. Asegúrate de que tu programa es robusto y por tanto comprueba que la letra leída es correcta, en caso contrario haz que se muestre un mensaje de error.
14. Escribe un programa que lea una serie de números enteros hasta que se introduzca el número -9999, y cuente el total de números introducidos, el total de valores positivos y el total de valores negativos (no consideres el cero ni positivo ni negativo). Reutilizar la función del ejercicio 9 para validar tus entradas.
15. Obtén las raíces de una ecuación de segundo grado  $ax^2 + bx + c = 0$  a partir de sus coeficientes a, b y c. Recuerda que si el discriminante es cero la raíz es única ( $-b/2a$ ) y que si el discriminante es negativo las raíces son imaginarias, en cuyo caso deberá indicarse con un mensaje "raíces imaginarias".
16. Escribe un programa que pida un número límite y calcule cuántos términos de la [serie armónica](#) son necesarios para que su suma supere dicho límite. Es decir, dado un límite se trata de determinar el menor número n tal que:

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} > \text{límite}$$

Pruebe su código, y tenga en cuenta que para valores altos del límite el tiempo de cálculo se dispara. Así, para un límite = 5, n sería 83; para 10 ya asciende a 12.367 y para 15, ¡¡el número de términos es 1.835.421!! El programa ha de ser

robusto, es decir, ha de controlar que el número introducido por el usuario es un entero positivo.

17. Escribe un programa que lea dos caracteres y averigüe si se introdujeron en el orden de su código ASCII o no. Se deberá además comprobar si el primero de ellos es una cifra, y en caso afirmativo indicar si es impar y si es o no primo. Modulariza la solución.
18. Escribe un programa que lea un entero positivo  $n$  y genere:
  - a. una tabla con las  $n$  primeras potencias de 2.
  - b. una tabla con las potencias de 2 que son menores o iguales que  $n$ .
19. Escribe un programa que acepte un año escrito en cifras arábicas y lo visualice escrito en números romanos, dentro del rango 1 a 2000. (I=1, V=5, X=10, L=50, C=100, D=500, M=1000).
20. Escribe un programa que muestre un menú en pantalla que permita calcular el seno, coseno, tangente, cotangente, secante y cosecante de un ángulo. El menú se mostrará hasta que el usuario decida salir. El menú quedará:

```
1. Seno
2. Coseno
...
0. Salir
Elija una opción:
```

21. El director de un banco necesita un programa para determinar cuándo un solicitante de un préstamo hipotecario está cualificado para obtenerlo. El criterio para la concesión del préstamo se guía por las siguientes características:
  - a. el solicitante debe haber trabajado al menos 5 años
  - b. su sueldo base debe ser al menos el 20% del precio de la vivienda
  - c. sus ingresos anuales han de ser al menos el 40% del precio de la compra
  - d. si tiene otros créditos, éstos no deben superar el 20% del sueldo anual

Realiza un programa modularizado para decidir si un solicitante es válido o no.

22. Una compañía de alquiler de automóviles desea un programa para emitir sus facturas, con las siguientes consideraciones:
  - a. Una cantidad fija de 100EUR si no se rebasan los 300 Km.
  - b. Si la distancia recorrida es mayor que 300 Km:
    - i. Si la distancia es menor o igual que 1000 Km, se cobrarán las 100EUR más el kilometraje que exceda de 300 Km a razón de 30 céntimos/Km.
    - ii. Si la distancia es mayor que 1000 Km, se cobrarán los 100EUR más el kilometraje a razón de 30 céntimos/Km para los kilómetros entre el 300 y el 1000 y 20 céntimos/Km para el resto.