

# PROGRAMACIÓN

1º CFGS DAM

Tema 01

Tipos de datos básicos  
(Continuación)

## Guía de ejercicios para la Unidad 01

1. Escriba un programa en Python que lea un número entero desde el teclado, y muestre por pantalla el número de dígitos que tiene. El programa debe verificar que la entrada sea correcta, es decir, que diga error si no se pone un número.

La ejecución del programa debe ser como la que se muestra a continuación:

```
Introduzca un número: -125
> El número -125 tiene tres dígitos
```

2. Escriba un programa en Python que lea un número (n) mayor o igual a 0 y muestre por pantalla los n primeros números de Fibonacci. El programa debe verificar que la entrada de datos sea correcta, es decir, no se permite letras y verificar que el número introducido es mayor o igual a 1.

La secuencia de Fibonacci es la que se define a continuación.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Donde  $F(0) = 0$ ,  $F(1) = 1$  y a partir de  $F(2)$  se calcula como la suma de los dos números anteriores.

$$F(n) = F(n-1) + F(n-2) \text{ con } n \geq 2$$

La ejecución del programa debe ser como la que se muestra a continuación:

```
Introduzca un número: 11
> 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
```

(se muestran 12 números porque se empieza con 0.

3. Modifique el programa anterior para que ahora en vez de mostrar los n primeros números de Fibonacci, muestre la suma de esos números.

La ejecución del programa debe ser como la que se muestra a continuación:

Introduzca un número: 11

> La suma de los 11 primeros números de Fibonacci es: 232

4. Escriba un programa en Python que lea un numero (n) mayor o igual a 0 y muestre por pantalla si el número leído es un número de Fibonacci. El programa debe verificar que la entrada de datos sea correcta, es decir, no se permite letras y verificar que el número introducido es mayor o igual a 0.

Ejemplos de la ejecución del programa:

Introduzca un número: 144

> El número 144 SI es un número de Fibonacci

Introduzca un número: 75

> El número 144 NO es un número de Fibonacci

5. Escriba un programa en Python que lea un numero (n) mayor o igual a 1 y muestre por pantalla las n primeras potencias de 2 El programa debe verificar que la entrada de datos sea correcta, es decir, no se permite letras y verificar que el número introducido es mayor o igual a 1.

La salida debe ser como la que se muestra a continuación:

Introduzca un número: 8

1

2

4

8

16

32

64

128

6. Modifique el programa anterior para que ahora en vez de mostrar los  $n$  primeras potencias de 2, muestre la suma de ellas.

La ejecución del programa debe ser como la que se muestra a continuación:

Introduzca un número: 8

> La suma de las 8 primeras potencias de 2 es: 255

7. Escriba un programa en Python que lea un número ( $n$ ) mayor o igual a 0 y muestre por pantalla si el número leído es una potencia de 2. El programa debe verificar que la entrada de datos sea correcta, es decir, no se permite letras y verificar que el número introducido es mayor o igual a 0.

Ejemplos de la ejecución del programa:

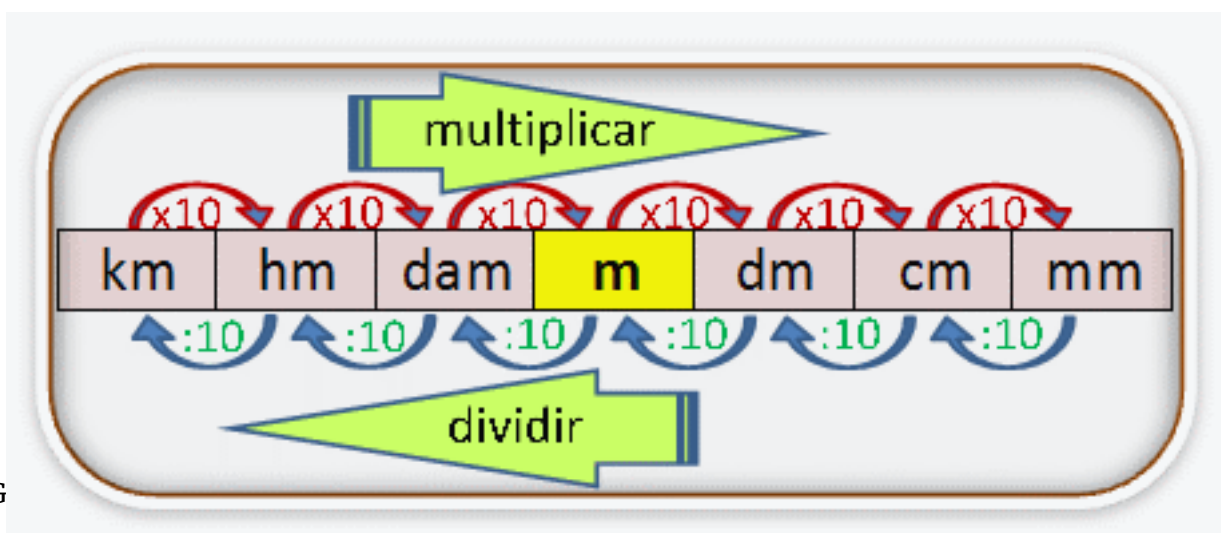
Introduzca un número: 128

> El número 128 SI es una potencia de 2

Introduzca un número: 75

> El número 75 NO es una potencia de 2

8. Escriba un programa en Python que permita convertir unidades de longitud en el sistema métrico decimal. Las unidades que se trabajarán son las que se muestran en la figura:



El programa debe leer:

- El valor de la medida que se quiere convertir (un número decimal)
- La unidad inicial y la unidad de destino en la misma línea separados por un carácter en blanco

El programa debe verificar todos los valores introducidos, es decir, no se permite letras y verificar que el número introducido es mayor o igual a 0..Los valores que se permiten en las unidades son los que se pueden ver en la figura: (km, hm, dam, m, dm, cm, mm)

Ejemplos de la ejecución del programa:

Introduzca un número: 128.75  
Unidad inicial y final separada por blanco: dam mm  
> 128.75 dam son 1287500 mm

Introduzca un número: 45,75  
Unidad inicial y final separada por blanco: dm hm  
> 45.75 dm son 0,4575 hm

9. Escriba un programa en Python que lea un número (n) mayor o igual a 0 y muestre por pantalla el resultado de calcular el factorial de número. El programa debe verificar todos los valores introducidos, es decir, no se permite letras y verificar que el número introducido es mayor o igual a 0.

El factorial de un número se calcula de la siguiente forma:

$$n! = n * (n-1) * (n-2) * \dots * 2 * 1. \text{ Con } 0! = 1$$

$$\text{Como ejemplo } 6! = 6 * 5 * 4 * 3 * 2 * 1 = 720$$

Ejemplos de la ejecución del programa:

Introduzca un número: 7  
> El factorial del número 6 es: 5040

10. Escriba un programa en Python que lea un número (n) entero y muestre por pantalla el que se obtiene al invertir sus dígitos. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que no contenga decimales.

La ejecución del programa debe ser como la que se muestra a continuación:

Introduzca un número: 734

> El número con las cifras invertidas de 734 es 437

11. Escriba un programa en Python que lea una frase y muestre por pantalla si es palíndromo o no

Una frase es palíndrome cuando lo que se lee de izquierda a derecha es lo mismo que lo que se lee de derecha a izquierda.

En este ejercicio es una buena idea quitar los espacios en blancos que puedan existir entre las palabras, así será mas fácil comprobar si la frase es igual o no.

Pueden usar los métodos explicados en clase para trabajar con string.

Se trata de comparar el string con su inverso. Es recomendable que pasen la frase a minúsculas o mayúsculas para hacer la comparación.

La ejecución del programa debe ser como la que se muestra a continuación:

Introduzca la frase: Dabale arroz a la zorra el abad

> La frase es PALÍNDROME

12. Escriba un programa en Python que lea un período de tiempo en días, horas, minutos y segundos y muestre por pantalla el número de segundos totales del período de tiempo indicado. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que correspondan al rango de cada una de las peticiones.

Ejemplo de la ejecución del programa:

Introduzca el número de días: 0

Introduzca el número de horas: 1

Introduzca el número de minutos: 1

Introduzca el número de segundos: 1

> El total de segundos del período indicado es: 3661

13. Realizar el mismo ejercicio anterior pero ahora el formato de entrada será: hh:dd:mm:ss. Se debe seguir verificando que todos los datos introducidos son correctos.

Ejemplo de la ejecución que debe tener el programa:

Introduzca el período de tiempo a convertir: 00:01:01:01

> El total de segundos del período indicado es: 3661

14. Escriba un programa en Python que lea un período de tiempo en segundos y muestre por pantalla el número de días, horas, minutos y segundos que representa. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que el número es mayor o igual a 0.

Ejemplo de la ejecución del programa:

Introduzca el número de segundos: 3661

> El período de tiempo es el siguiente:

> 0 días, 0 meses, 1 hora, 1 minuto, 1 segundo

15. Escriba un programa en Python que lea un número (n) entero positivo y muestre por pantalla todos los divisores que tiene. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que el número es mayor o igual a 0.

Ejemplo de la ejecución del programa:

Introduzca el número de segundos: 32

> Los divisores del número 32 son: 1, 2, 4, 8, 16 y 32

16. Escriba un programa en Python que lea un número (n) entero positivo y muestre por pantalla los 20 primeros múltiplos del número leído. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que el número es mayor o igual a 0. La salida debe ser tabulada y se debe escribir 4 filas de 5 múltiplos cada una.

Ejemplo de la ejecución del programa:

3	6	9	12	15
18	21	24	27	30
33	36	39	42	45
48	51	54	57	60

17. Escriba un programa en Python que lea una frase por pantalla y escriba el número de vocales que contiene.

Ejemplo de la ejecución del programa:

Introduzca una frase: amanecer

> La frase amanecer tiene 4 vocales

18. Modifique el programa anterior para que clasifique las vocales y diga cuántas hay de cada una de ellas.

Ejemplo de la ejecución del programa:

Introduzca una frase: amanecer

> La frase amanecer tiene :

> 2 a

> 2 e

> 0 i

> 0 o

> 0 u



19. Escribir un programa que lea un string en pantalla que represente la dirección de un correo electrónico y diga si es válida o no. Lo que se pide es que se revise el formato.

Una dirección de correo debe tener el siguiente formato: `aaaaaaaa@aaaaaaaa.aaa`. Es decir, **antes del arroba debe existir al menos un carácter y no debe empezar por número**, a continuación el carácter '@', seguido de cualquier número de caracteres o dígitos, (que no comienza por dígito), para finalizar debe estar el '.' seguido de al menos 2 y hasta 4 caracteres, el primero de los cuales no puede ser dígito.

Una dirección correcta es : `regonher@canariaseducacion.es`

20. Escriba un programa en Python que lea un número (n) entero positivo y muestre por pantalla la secuencia de Collatz. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que el número es mayor a 0.

El procedimiento a realizar para obtener la secuencia de Collatz es el siguiente:

- Si el número es par, se debe dividir por 2.
- Si es impar, se debe multiplicar por 3 y sumarle 1.

Con esto se obtiene el siguiente número de la sucesión, al cual se le deben aplicar las mismas operaciones. La sucesión de números termina cuando el número obtenido por medio de las operaciones de 1.

Introduzca el número : 6

> La secuencia de Collatz para el número 6 es:

> 6 3 10 5 16 8 4 2 1

21. Escriba un programa en Python que lea un número (n) entero positivo y muestre si el número es mágico o no. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que el número es mayor a 0. En este caso un número se considera mágico cuando la suma de sus divisores sin contar el propio número es igual al número.

Ejemplos de la salida que debe mostrar el programa:

Introduzca un número: 28.                       $28 = 1 + 2 + 4 + 7 + 14$   
> 28 es número mágico

Introduzca un número: 32                       $32 \neq 1 + 2 + 4 + 8 + 16$   
> 32 NO es número mágico

22. Escribir un programa en Python que genere un número entre 1 y 10, y luego de 5 opciones de poder adivinarlo. Si se adivina, debe mostrarse el número y decir en cuántos intentos se ha realizado. Si no se adivina se escribirá un mensaje que no se ha adivinado el número y se escribirá el mismo.

Para este programa es necesario el uso del generador de números aleatorios. Para ello se debe importar la librería random y la función a utilizar es:

*randint(valor\_inicial, valor\_final),*

que genera un número aleatorio entre [valor\_inicial, valor\_final].

Ejemplos de la salida que debe mostrar el programa:

Suponiendo que el número generado fue el 8:

Introduzca número del intento 1: 2  
Introduzca número del intento 2: 1  
Introduzca número del intento 3: 9  
Introduzca número del intento 4: 7  
Introduzca número del intento 5: 6

Ohhhh... no has adivinado el número. Este era el 8.  
Suponiendo que el número generado fue el 3:

Introduzca número del intento 1: 5

Introduzca número del intento 2: 3

Enhorabuena...has adivinado el número en 2 intentos.

23. Escriba un programa en Python que lea un número (n) entero positivo y muestre como salida el resultado de sumar los cuadrados de los números desde el 1 hasta el valor leído. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que el número es mayor a 0.

Ejemplos de la salida que debe mostrar el programa:

Introduzca un número: 4

> El resultado de la suma de los cuadrados es: 30

24. Escriba un programa en Python que lea un número (n) entero positivo que indicará la cantidad de números a leer posteriormente. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que el número es mayor a 0. Luego se leen esa cantidad de números y se debe mostrar el mayor y el menor de ellos.
25. Modificar el programa anterior para que muestre además la cantidad de veces que han salido.
26. Escriba un programa en Python que muestre todas las combinaciones posibles que pueden salir al lanzar dos dados. De cada combinación sume el número de puntos obtenidos y calcule la probabilidad de que salga mostrando 4 decimales.

La salida debe ser como la siguiente:

1 - 1 total de puntos 2

2 - 1 total de puntos 3

3 - 1 total de puntos 4

- .
- .
- 6 - 6 total de puntos 12
- La probabilidad de que la suma de 2 es = 0,0278
- La probabilidad de que la suma de 3 es = 0,0556
- .
- .
- La probabilidad de que la suma de 12 es = 0,0278

27. Escriba un programa en Python que lea un número (n) entero positivo y diga si es primo o no. El programa debe verificar que el dato leído es correcto, es decir, no se permite letras y que el número es mayor a 0. Un número es primo cuando sólo es divisible por 1 y por él mismo.
28. Modificar el algoritmo anterior para que muestre los números primos que hay entre un rango de números solicitado por pantalla.

Por ejemplo, si se lee, 2 y 15

El programa debe escribir: 2, 3, 5, 7, 11, 13

29. Escribir un programa en Python que lea dos números enteros a y b y calcule el MCD y el mcm de ellos. El programa debe verificar que tanto a como b son correctos, es decir, no se permite letras y que el número es mayor a 0.

Este programa se debe realizar en dos versiones:

La primera y más fácil es importando la librería math de Pythpmn que ya implementa funciones para su cálculo. Averigüe cuales son y aplíquelas.

La segunda versión debe calcularse sin el uso de esas funciones, para ello averigüe en internet como se debe hacer para calcular los dos valores.

La salida debe ser como la que se muestra a continuación:

MCD(a,b) = resultado.    mcm(a,b) = resultado

30. Realice un programa en Python que pida los coeficientes de una ecuación de segundo grado y muestre los resultados de la soluciones reales si las tiene. El programa debe verificar que los valores leídos para a, b y c son números. El programa debe verificar si la ecuación dada tiene solución real y mostrarla por pantalla, en caso contrario debe indicar que no tiene solución real.

Recuerden que la ecuación de segundo grado se escribe de forma genérica:  $a * x^2 + b * x + c = 0$  que tiene las siguientes soluciones:

$$x1 = -b + (b^2 - 4*a*c) / 2 * a$$

$$x2 = -b - (b^2 - 4*a*c) / 2 * a$$

31. Realice un programa en Python que encuentre los números naturales menores de 1000 en los que la suma de los cubos de sus dígitos sea igual al propio número.

Ejemplo:

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

32. Año bisiesto. Cuando la Tierra completa una órbita alrededor del Sol, no han transcurrido exactamente 365 rotaciones sobre sí misma, sino un poco más. Más precisamente, la diferencia es de más o menos un cuarto de día. Para evitar que las estaciones se desfasen con el calendario, el calendario juliano introdujo la regla de introducir un día adicional en los años divisibles por 4 (llamados bisiestos), para tomar en consideración los cuatro cuartos de día acumulados. Sin embargo, bajo esta regla sigue habiendo un desfase, que es de aproximadamente 3/400 de día. Para corregir este desfase, en el año 1582 el papa Gregorio XIII introdujo un nuevo calendario, en el que el último año de cada siglo dejaba de ser bisiesto, a no ser que fuera divisible por 400.

Escriba un programa en Python que pida un número que representa un año e indique si un año es bisiesto o no. El programa debe verificar que

el dato leído es correcto, es decir, no se permite letras y que el número es mayor a 0.

La ejecución del programa debe ser la siguiente:

Introduzca un año: 2020

> El año 2020 SI es bisiesto.

33. Escriba un programa en Python que pida al usuario dos palabras, y que indique cual de ellas es la más larga, y muestre el trozo de string por el que difieren en longitud.

La salida debe ser como la siguiente:

Introduzca una palabra: sandía

Introduzca una segunda palabra: zanahoria

>zanahoria tiene 9 letras

>sandía tiene 6 letras

> por tanto zanahoria es más larga

> el string que difieren en longitud : ria

34. Escriba un programa en Python que pida al usuario un número decimal y muestre por pantalla la parte decimal del mismo. El programa debe verificar que se introduce un número correcto. Un ejemplo de la ejecución del programa sería:

Introduzca un número decimal: -2,25

> La parte decimal del número es 0,25

35. Escriba un programa en Python que pida al usuario dos números enteros o decimales que representan las dimensiones de los catetos de un triángulo rectángulo y muestre por pantalla la longitud de su hipotenusa. Además se debe mostrar el área y el perímetro del triángulo. El programa debe verificar que la entrada es correcta, y que los números introducidos son positivos.