

**Grado en Inteligencia Artificial**  
**Programación I**  
**Práctica 4**

1.- Define una **función** que convierta grados Fahrenheit en grados centígrados. Para calcular los grados centígrados has de restar 32 a los grados Fahrenheit y multiplicar el resultado por cinco novenos. Define otra función que convierta grados centígrados en grados Fahrenheit.

2.- Define una **función** que convierta grados en radianes. Define una función que convierta radianes en grados. Recuerda que 360 grados son  $2\pi$  radianes.

3.- Define una **función** que devuelva el número de días que tiene un año determinado. Un año es bisiesto si es divisible por 4 y no divisible por 100, excepto si es también divisible por 400, en cuyo caso es bisiesto.

Ejemplos: El número de días de 2002 es 365: el número 2002 no es divisible por 4, así que no es bisiesto. El año 2004 es bisiesto y tiene 366 días: el número 2004 es divisible por 4, pero no por 100, así que es bisiesto. El año 1900 es divisible por 4, pero no es bisiesto porque es divisible por 100 y no por 400. El año 2000 sí es bisiesto: el número 2000 es divisible por 4 y, aunque es divisible por 100, también lo es por 400.

4.- Diseña una **función** que calcule  $\sum_a^b i$  dados  $a$  y  $b$ . Si  $a$  es mayor que  $b$ , la función devolverá el valor 0.

5.- Diseña una **función** que calcule  $\prod_a^b i$  dados  $a$  y  $b$ . Si  $a$  es mayor que  $b$ , la función devolverá el valor 0. Si 0 se encuentra entre  $a$  y  $b$ , la función devolverá también el valor cero, pero sin necesidad de iterar en un bucle.

6.- Define una **función** llamada `raizEnesima` que devuelva el valor de  $\sqrt[n]{x}$ .

7.- Diseña una **función** que diga (mediante la devolución de `True` o `False`) si dos números son amigos. Dos números son amigos si la suma de los divisores del primero (excluido él) es igual al segundo y viceversa.

8.- Diseña una **función** sin argumentos que devuelva un número aleatorio mayor o igual que 0.0 y menor que 10.0. Puedes llamar a la función `random` desde tu función. Diseña una función sin argumentos que devuelva un número aleatorio mayor o igual que -10.0 y menor que 10.0.

9.- Para diseñar un juego de tablero nos vendrá bien disponer de un “dado electrónico”. Escribe una **función** Python sin argumentos llamada `dado` que devuelva un número entero aleatorio entre 1 y 6.

10.- Diseña un programa que, dado un número  $n$ , muestre por pantalla todas las parejas de números amigos menores que  $n$ . La impresión de los resultados debe hacerse desde un **procedimiento**. Dos números amigos solo deberán aparecer una vez por pantalla. Por ejemplo, 220 y 284 son amigos: si aparece el mensaje «220 y 284 son amigos», no podrá aparecer el mensaje «284 y 220 son amigos», pues es redundante. Debes diseñar una función que diga si dos números son amigos y un procedimiento que muestre la tabla.

11.- Se llaman números primos gemelos a los pares de números primos que son impares consecutivos (3 y 5, 11 y 13, ...). Construye un programa que imprima los números gemelos que hay entre 1 y un número que se le pide al usuario del programa. [Primer parcial 2019]

12.- Una torre en ajedrez se mueve en línea recta a través de las filas y columnas. Supón un tablero vacío en el que solamente hay una torre en una casilla origen que tiene que llegar a una casilla destino que no está en la misma fila y columna que la casilla de origen. Hay muchos caminos posibles, pero solamente dos que se puedan realizar con dos movimientos. Se pide hacer un programa en Python tal que, dadas las coordenadas de las casillas de origen y destino, muestre las coordenadas de las casillas visitadas por la torre en esos dos caminos. [Primer parcial 2019]

13.- En ajedrez, el alfil se mueve sobre el tablero en una línea recta diagonal. Se puede mover tantas casillas como se quiera, hasta que se encuentre con el final del tablero o con otra pieza. Supongamos que las casillas de ajedrez se pueden referir mediante un número de fila y de columna, del 1 al 8. Se pide hacer un programa que pida al usuario un número de fila y columna, compruebe que son válidos y que imprima por pantalla la serie de casillas accesibles por el alfil situado en dicha casilla, suponiendo que no hay más piezas en el tablero. En el ejemplo de la Figura 1, el usuario habría introducido la casilla inicial de la fila 5 y la columna 5, y el programa imprimiría las coordenadas de las casillas accesibles, marcadas con una cruz verde. [Primer parcial 2018]

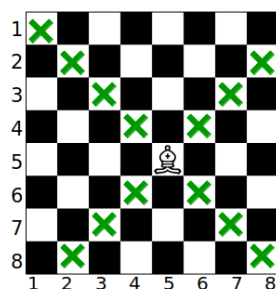


Figura 1: Movimientos del alfil

14.- Escribe un programa que pida al usuario un número entero mayor que 3 y que muestre por pantalla su descomposición en factores primos. Por ejemplo, si la entrada es 60, deberá imprimir 2, 2, 3, 5 en líneas separadas. [Primer parcial 2017]

15.- Implementa un **procedimiento** Python tal que, dado un número entero, muestre por pantalla sus cifras en orden inverso. Por ejemplo, si el procedimiento recibe el número 324, mostrará por pantalla el 4, el 2 y el 3 en líneas diferentes.

16.- Diseña una **función** esPrimo que determine si un número es primo (devolviendo True) o no (devolviendo False). Diseña a continuación un **procedimiento** muestraPrimos que reciba un número y muestre por pantalla todos los números primos entre 1 y dicho número.

17- Diseña una función recursiva que calcule el número de dígitos que tiene un número entero en base 10.

18.- Puedes calcular recursivamente los números combinatorios sabiendo que, para  $n \geq m$ ,

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}$$

y que

$$\binom{n}{n} = \binom{n}{0} = 1$$

Diseña un programa que, a partir de un valor  $n$  leído de teclado, muestre  $\binom{n}{m}$  para  $m$  entre 0 y  $n$ . El programa llamará a una función de nombre *combinaciones* definida recursivamente.

19.- Diseña una función recursiva que calcule la suma de los  $n$  primeros números naturales.

20.- Diseña un procedimiento recursivo que imprima la lista de números naturales comprendidos entre dos valores *ini* y *fin* dados por el usuario.

21.- Diseña una función recursiva que calcule  $x^y$  mediante multiplicaciones sucesivas, siendo  $x$  e  $y$  dos números enteros.

22.- Diseña una función recursiva que calcule  $x*y$  mediante sumas sucesivas, siendo  $x$  e  $y$  dos números enteros.

23.- Desarrolla un programa que reciba un número entero de teclado. A continuación, deberá: (i) Calcular mediante un algoritmo recursivo el número de cifras que tiene dicho número y, (ii) utilizando este dato y el número de partida, dar la vuelta al número mediante otro algoritmo recursivo. Finalmente, el programa imprimirá por pantalla el número de cifras y el número dado la vuelta. [**Primer parcial 2019**]

24.- Diseñe una función **recursiva** (serieRec) y otra **iterativa** (serieIter) que permita calcular la suma de la siguiente serie:

$$\sum_{i=1}^n \frac{1}{i}$$

donde  $n \geq 1$  es un número entero que se recibe como argumento de la función. [**Primer parcial 2018**]

25.- Diseña una función **recursiva** (sumaCifrasR) y otra **iterativa** (sumaCifrasI) que permitan sumar los dígitos de un número. Por ejemplo, si la entrada es 9999 el resultado que devolverá es 36. [**Primer parcial 2017**]