

1) Escribir un programa que sea capaz de resolver sistemas de ecuaciones lineales con dos incógnitas, es decir, sistemas de la forma:

$$ax + by = c$$

$$dx + ey = f$$

Nota. El programa deberá tener en cuenta los casos en los que el sistema no tenga solución.

2) Dada la siguiente función:

$$f(x, y) = \begin{cases} 10 & \text{si } x \times y < 1 \\ y^2 & \text{si } x \times y \geq 1 \end{cases}$$

Escribir un programa que calcule el valor de la función para los valores x e y solicitados al usuario.

3) Realizar un programa que imprima por pantalla los números primos que existen entre dos números solicitados al usuario. Si no existe ninguno, el programa informará de este hecho al usuario.

4) Realizar un programa que detecte si el número introducido por el usuario es un número capicúa.

Nota. Los números capicúas son aquellos que son simétricos, por ejemplo: 123321 o 121.

5) Determinar si un número entero positivo es perfecto.

Nota. Un número perfecto es aquel que la suma de sus divisores menos él mismo es el propio número. Los 5 primeros números perfectos son 6, 28, 496, 8128 y 33550336.

$$6 \text{ (divisores)} \Rightarrow 1 + 2 + 3 + 6 - 6 = 6$$

$$28 \text{ (divisores)} \Rightarrow 1 + 2 + 4 + 7 + 14 + 28 - 28 = 28$$

6) Escribir un programa que calcule el producto de los dígitos de un número entero solicitado al usuario. Tendrán que tenerse en cuenta también los números negativos.

7) El cálculo del valor de e viene dado por la expresión:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

Escribir un programa que calcule el valor de e. La condición de parada será la precisión indicada por el usuario mediante el número de términos de la serie que se desean calcular. En el ejemplo que se muestra tendríamos 4 términos.

8) Realizar un programa que obtenga el mcd de dos números enteros utilizando el algoritmo de Euclides. Dados A y B, ($A, B > 0$ y $A > B$), si el resto de la división entre A y B es cero, B es el máximo común divisor. Si el resto no es cero, el divisor y el resto sirven respectivamente de dividendo y divisor de la siguiente división. El proceso termina cuando se obtiene un resto igual a cero. El mcd es entonces el penúltimo resto calculado.

Ejemplo: A = 1498 y B = 980

1498 % 980 = 518

980 % 518 = 462

518 % 462 = 56

462 % 56 = 14

56 % 14 = 0

El máximo común divisor es 14.

9) Determinar si dos números enteros positivos son amigos.

Nota. Dos números a y b son amigos si a es la suma de los divisores de b, y b es la suma de los divisores de a. El uno se considera divisor, pero no el propio número.

Ejemplo: 220 y 284.

220 (divisores) _ 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284

284 (divisores) _ 1 + 2 + 4 + 71 + 142 = 220

10) Crear un programa que calcule el valor de la intensidad que pasa a través de una resistencia dada, cuando se le aplica un voltaje determinado. El programa deberá estar dividido en las siguientes funciones:

a. explicar_programa(). Esta función mostrará una introducción del programa por la pantalla.

b. obtener_valores(). Esta función pedirá los valores para la resistencia y voltaje, los cuales se pasarán por referencia al programa principal.

c. calcular(). Esta función efectuará el cálculo de la intensidad a partir de la resistencia y el voltaje aplicado.

d. imprimir_respuesta(). Esta función se encargará de mostrar un mensaje con los resultados.

11) Escribir un programa que calcule el número combinatorio:

$$\binom{M}{N} = \frac{M!}{N!(M - N)!}$$

12) Realizar un programa que pase un número binario (parte entera y parte fraccionaria) a decimal. Para ello, deberá pedir al usuario que indique cuántos bits de la parte entera del número va a introducir y cuantos de la parte fraccionaria. Posteriormente, solicitará los bits de la parte entera comenzando por el más significativo (el de mayor peso). Una vez haya obtenido el valor decimal de la parte entera, solicitará los bits correspondientes a la parte fraccional del número binario, comenzando de nuevo por el de mayor peso. Finalmente, mostrará por pantalla el valor decimal del número binario introducido.

13) Determinar el mínimo común múltiplo (m.c.m) de dos números enteros. Los múltiplos de un número se obtienen multiplicando dicho número por los números naturales.

Nota. El m.c.m de dos números es el menor múltiplo común distinto de cero, o dicho de otra forma, el m.c.m de dos números expresados como producto de factores primos, su mínimo común múltiplo será el resultado de multiplicar los factores comunes y no comunes elevados al mayor exponente:

Ejemplo: El m.c.m de 10 y de 20 es 20. El m.c.m de 4 y 5 es 20.

14) Realizar un programa que lea en la función principal las coordenadas de un punto (x,y) y que mediante un menú invoque la siguientes funciones:

- Comprobar si el punto pertenece a una circunferencia de radio r y centro en el punto (x₂,y₂), valores solicitados al usuario.
- Determinar en qué cuadrante se encuentra el punto dado. Se deberá prestar especial atención a los puntos que pertenecen a los ejes de coordenadas, o cuando se trate del punto (0,0), mediante mensajes informativos mostrados por pantalla.
- Convertir las coordenadas del punto (x,y) a coordenadas polares (radio,ángulo). La función en este caso tendrá cuatro parámetros. La función principal será la encargada de imprimir en pantalla el resultado de la ejecución de esta función.

También deberá mostrarse un opción al usuario para que abandone el programa cuando desee.

15) Realizar un programa que, mediante un menú e implementadas mediante funciones, permita realizar las siguientes operaciones con polinomios de segundo grado:

- Leer un polinomio de segundo grado.
- Imprimir el polinomio de segundo grado que haya sido leído en última instancia.
- Calcular el valor del polinomio $f(x)$ en un punto x , cuando x toma un valor real.
- Calcular la derivada del polinomio de segundo grado introducido.

El usuario podrá abandonar el programa cuando desee, reflejando dicho hecho mediante una opción en el menú.

16) Realizar un programa que, mediante un menú e implementadas mediante funciones, permita realizar la suma, resta, multiplicación, y división de dos números complejos cualesquiera introducidos por el usuario. También se podrá realizar el producto escalar y valor absoluto de uno de ellos, operaciones implementadas también mediante funciones. El usuario podrá introducir el valor de los números complejos con los que desea trabajar mediante opciones en el menú.

Finalmente, dicho usuario podrá abandonar el programa mediante una de las opciones del menú.

17) Escribir un programa que calcule el número de billetes de 500, 200, 100, 50, 20, 10 y 5 euros, así como de monedas de 2 y 1 euros, y de 50, 20, 10, 5, 2 y 1 céntimos de euros; para desglosar una cantidad dada, de forma que se necesite la menor cantidad de monedas y billetes de cada tipo. Ejemplo: 535,45 es 1 billete de 500, 1 billete de 20 euros, 1 billete de 10 euros, un billete de 5 euros, dos monedas de 20 céntimos de euro y una moneda de 5 céntimos de euro.

18) Dada una cadena cualquiera, realizar un programa que extraiga los números que aparecen en dicha secuencia, que los transforme en números enteros y los imprima por pantalla.

19) Realizar un programa que indique si una cadena de caracteres introducida por teclado es un palíndromo (un palíndromo es una cadena que se lee igual de izquierda a derecha que de derecha a izquierda, como por ejemplo “dabale arroz a la zorra el abad” o “alolocolocololola”).

20) Realizar un programa que declare e inicialice una matriz cuadrada y que calcule e imprima su traspuesta. Imprimir la matriz original y la traspuesta

21) Dado un vector de números enteros de cualquier longitud declarado en el programa principal, escribir una función que ordene los elementos del vector de tal forma que los números pares aparezcan antes que los números impares. Además, los números pares deberán estar ordenados de forma ascendente, mientras que los números impares deberán estar ordenados de forma descendente. Por ejemplo, el vector {1,2,3,4,5,6} quedará como {2,4,6,5,3,1}.

22) Escribir un programa que calcule el producto escalar y vectorial de dos vectores de 3 elementos, cuyos valores serán solicitados al usuario.

23) Escribir un programa que escriba en cada posición de una matriz 10x10 el valor de la distancia euclídea de dicha posición al elemento de la esquina inferior izquierda. Mostrar por pantalla el resultado. Considérese que el valor $a_{i,j}$ ocupa la posición (i,j).
Nota. La distancia euclídea entre dos puntos (a,b) y (c,d) puede definirse como:

$$d = \sqrt{(c - a)^2 + (d - b)^2}$$

24) El cifrado César o código César, es una de las técnicas de codificación más simples y más usadas que se conoce. Es un tipo de cifrado por sustitución, en el que cada letra del texto es reemplazada por otra letra que se encuentra un número fijo de posiciones más adelante en el alfabeto. Por ejemplo, con un desplazamiento de 3, la A sería sustituida por la D (situada 3 lugares a la derecha de la A), la B sería reemplazada por la E, y así sucesivamente. Este método debe su nombre a Julio César, ya que lo usaba para comunicarse con sus generales.

Implementar una función que permita codificar cadenas de caracteres mediante códigos Cesar de cualquier longitud. La función principal deberá hacer uso de dicha función, solicitando al usuario tanto la cadena de caracteres a codificar, como el valor del desplazamiento a emplear.

Nota. Emplear sólo desplazamientos de los caracteres del alfabeto recogidos en la Tabla ASCII Extendida. Los caracteres ñ y Ñ se dejarán sin desplazar.

25) Dadas dos matrices 4x4 de números enteros declaradas en la función principal, se pide realizar un programa que permita mediante opciones acceder a las distintas funciones que a continuación se detallan:

- Escribir una función *lectura()*, que permita al usuario introducir los valores a almacenar en cualquiera de las dos matrices.
- Escribir una función *escribir()*, que permita al usuario modificar el valor de una posición en concreto de cualquiera de las dos matrices. La matriz, la posición a modificar y el nuevo valor serán pasados a la función mediante parámetros.
- Escribir una función *imprimir()*, que permita imprimir los valores almacenados en cualquiera de las dos matrices.
- Escribir una función *borrar()* que nos permita borrar todos los valores almacenados en cualquiera de las dos matrices. **Nota.** Esta función deberá escribir en todas las posiciones de la matriz el valor 0.
- Escribir una función *suma()* que realice la suma de las dos matrices, almacenando el resultado en una de ellas.
- Escribir una función *multiplica()* que realice la multiplicación de las dos matrices, almacenando el resultado en una de ellas.
- Escribir una función *traspuesta()* que obtenga la traspuesta de cualquiera de las dos matrices, almacenando el resultado en la matriz original pasada como parámetro.



SISTEMAS DE GESTIÓN EMPRESARIAL
2º CFGS DAM
EJERCICIOS PYTHON 2



26) (Opcional) **Problema de dificultad adicional.** Utilizando los recursos al alcance interpretar la codificación-decodificación Vignère e implementar un algoritmo para poder cifrar y descifrar un texto.