

1

Solicitar al usuario que ingrese su dirección email. Imprimir un mensaje indicando si la dirección es válida o no, valiéndose de una función para decidirlo. Una dirección se considerará válida si contiene el símbolo "@".

2

Solicitar números al usuario hasta que ingrese el cero. Por cada uno, mostrar la suma de sus dígitos (utilizando una función que realice dicha suma).

3

Solicitar números al usuario hasta que ingrese el cero. Por cada uno, mostrar la suma de sus dígitos. Al finalizar, mostrar la sumatoria de todos los números ingresados y la suma de sus dígitos. Reutilizar la misma función realizada en el ejercicio 2.

4

Requerir al usuario que ingrese un número entero e informar si es primo o no, utilizando una función booleana que lo decida.

5

Solicitar al usuario un número entero y luego un dígito. Informar la cantidad de ocurrencias del dígito en el número, utilizando para ello una función que calcule la frecuencia.

6

Escribir un programa que pida números al usuario, mostrar el factorial de cada uno y, al finalizar, la cantidad total de números leídos en total. Utilizar una o más funciones, según sea necesario.

7

Escribir un programa que pida números positivos al usuario. Mostrar el número cuya sumatoria de dígitos fue mayor y la cantidad de números cuya sumatoria de dígitos fue menor que 10. Utilizar una o más funciones, según sea necesario.

8

Solicitar al usuario el ingreso de números primos. La lectura finalizará cuando ingrese un número que no sea primo. Por cada número, mostrar la suma de sus

dígitos. También solicitar al usuario un dígito e informar la cantidad de veces que aparece en el número (frecuencia). Al finalizar el programa, mostrar el factorial del mayor número ingresado.

9

Sin ejecutar el siguiente programa, determinar cuál es la salida en pantalla si se ingresan los valores $x=6$, $y=7$:

```
def coordenadaZ(x,y):
    x=x+10
    y=y+15
    return x+y

#programa principal
x=int(input("Coordenada eje x: "))
y=int(input("Coordenada eje y: "))
for i in range(3):
    z=coordenadaZ(x,y)
    x=x+1
    y=y+1
    print(x," . ",y)
```

10

El siguiente programa debería imprimir el número 2 si se le ingresan como valores $x=5$, $y=1$ pero en su lugar imprime 5. ¿Qué hay que corregir?

```
def maximo(a,b):
    if x>y:
        return x
    else:
        return y

def minimo(a,b):
    if x<y:
        return x
    else:
        return y

#programa principal
x=int(input("Un número: "))
y=int(input("Otro número: "))
print(maximo(x-3, minimo(x+2, y-5)))
```

11

Escribir una función que, dado un número de DNI, retorne True si el número es válido y False si no lo es. Para que un número de DNI sea válido debe tener entre 7 y 8 dígitos.

12

Escribir una función que, dado un string, retorne la longitud de la última palabra. Se considera que las palabras están separadas por uno o más espacios. También podría haber espacios al principio o al final del string pasado por parámetro.

13

Escribir un programa que permita al usuario obtener un identificador para cada uno de los socios de un club. Para eso ingresará nombre completo y número de DNI de cada socio, indicando que finalizará el procesamiento mediante el ingreso de un nombre vacío.

Precondición: el formato del nombre de los socios será: nombre apellido. Podría ingresarse más de un nombre, en cuyo caso será: nombre1 nombre2 apellido. Si un socio tuviera más de un apellido, el usuario sólo ingresará uno.

Se debe validar que el número de DNI tenga 7 u 8 dígitos. En caso contrario, el programa debe dejar al usuario en un bucle hasta que ingrese un DNI correcto. Por cada socio se debe imprimir su identificador único, el cual estará formado por: el primer nombre, la cantidad de letras del apellido y los primeros 3 dígitos de su DNI. Ejemplo:

Nombre: Alba María Linares

DNI: 25834910

Alba7258

14

Escribir la función titulo(), la cual recibe un string y lo retorna convirtiendo la primera letra de cada palabra a mayúscula y las demás letras a minúscula, dejando inalterados los demás caracteres. Precondición: el separador de palabras es el espacio: " ". Agregar doctests con suficientes casos de prueba para validar que la función retorna el valor esperado ante distintos argumentos.