

# Tarea programada #2 Valor 20%:

---

Cree las siguientes funciones y realice un llamado de ejemplo para cada una de ellas.

Todos los llamados de las funciones deben hacerse juntos y al final del archivo, es decir, luego de que todas las funciones fueron definidas.

---

1. Una función que reciba tres parámetros obligatorios:

El primero será un string con una de las siguientes operaciones:

- suma
- resta
- multiplicacion
- módulo

Los siguientes dos parámetros serán dos números enteros a los que se le aplicará la operación indicada.

Algunos ejemplos:

Si la función es llamada con los parámetros **"multiplicacion", 8, 3** el resultado devuelto debería ser **24**.

---

2. Una función que reciba una lista de números y devuelva su mediana:

Si se envía la siguiente lista **[1, 2, 3, 4, 5, 6, 7]** el resultado esperado es **4**.

---

3. Una función que reciba una cantidad cualquiera de elementos y devuelva todos los elementos concatenados separados por un espacio:

Si se le envían los siguientes parámetros : **"Este es el mes", 4, "=", True** debería devolver:

**"Este es el mes 4 = True"**

---

4. Una funcion, una que reciba los siguientes parámetros con las siguientes restricciones:

- nombre : requerido
- apellidos : requerido
- direccion : opcional
- edad : requerido
- profesion : opcional
- correo\_electronico : requerido
- telefono : opcional

Con esos parámetros debe devolver un diccionario con los datos de la siguiente forma:

```
{  
    "Nombre" : "Juan",  
    "Apellidos" : "Salas",  
    "Direccion" : "No indica",  
    "Edad" : 33,  
    "Profesión" : "Informático",  
    "Email" : "correo@ejemplo.com",  
    "Telefono" : "No indica",  
}
```

Del ejemplo anterior solo puede variar el valor de las entradas, las llaves deben respetarse. Si alguno de los valores opcionales no se indica, en el diccionario debe venir la frase **"No indica"**

- 
5. Una función que reciba dos valores, y devuelva el resultado de aplicar el operador + entre ambos datos. Si el operador no se puede utilizar entre los datos enviados, debe devolver un mensaje que diga **"No se puede aplicar este operador a los datos enviados"**.
-

6. Dos funciones que calculen el factorial de cualquier número entero mayor igual a cero y menor que 25. Una de las función debe resolverlo de forma recursiva y el otro de forma iterativa (opcional).

El factorial de un número se define como la multiplicación del número por todos los números anteriores a él. Así:

$$7! = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 5040$$

$$2! = 2 \times 1 = 2$$

Tome en cuenta que

$$0! = 1$$

---

7. Cree una función que reciba dos parámetros. Cada uno de los parámetros puede ser:

- piedra
- papel
- tijeras

Comparando los parámetros, el programa debe indicar si ganó el jugador 1 ( parámetro1) o el jugador2( parámetro 2)

---

*\*Opcional: \**

Cree un programa que permita a un jugador jugar piedra, papel o tijeras contra la máquina. El programa le permitirá seleccionar cual opción quiere utilizar el usuario (piedra, papel o tijeras), y cuando el usuario seleccione una el programa deberá seleccionar una para la máquina de forma aleatoria.

Se debe verificar al ganador y sumar un punto ya sea al usuario o a la máquina.

Al iniciar el programa, se debe solicitar la cantidad de turnos que se jugarán, que siempre debe ser un número impar mayor a cero( Si el número es par, no se debería volver a preguntar ). El ganador del juego es el que tenga más puntos al terminar la cantidad de selecciones indicadas. Cada vez que hay un ganador, se rebaja un turno, si no hay ganadores, no se rebaja turno.

Puede solicitar el nombre del usuario al inicio del programa para mostrarlo en la terminal.

Después de cada selección debería mostrarse quien ganó y el puntaje, similar a los siguiente.

=====

| Piedra papel tijeras | Turnos restantes: 5 |

=====

| Marcador: |

=====

| Maquina : [ 0 ]                      | Juan [ 2 ]                      |

=====

Este ejercicio se debe enviar en un documento adicional.

---

Se debe comentar cada una de las funciones de forma interna utilizando la siguiente plantilla en la línea siguiente a la firma de la función:

```
def abc () :  
    """  
  
    Descripción corta de la función  
  
    @nombre del parámetro 1 : Para que sirve el parámetro 1  
  
    @nombre del parámetro 2 : Para que sirve el parámetro 2  
  
    ...  
  
    @nombre del parámetro n : Para que sirve el parámetro n  
  
    @return : Indicar que retorna la función  
  
    """
```