

## PROBLEMAS PC3

### Problema 1:

Escriba 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.

### Problema 2:

Escriba una función 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: " ".

### Problema 3:

Escriba una función que acepte una secuencia de palabras separadas por guiones como entrada e imprima las palabras en una secuencia separada por guiones después de ordenarlas alfabéticamente.

### Problema 4:

Escriba una función que acepte dos listas y devuelva True si los elementos de una lista son cuadrados de los elementos de la otra lista e independientemente del orden, en caso contrario devuelva False.

Ejemplo:

a = [121, 144, 19, 3, 19, 144, 19, 11]

b = [121, 14641, 20736, 361, 9, 361, 20736, 361]

Resultado esperado: True

### Problema 5:

Escriba una función que reciba un string, la función debe devolver un substring que cumpla las siguientes condiciones:

- El substring debe ser la más larga de todas las posibles substrings en el string dado.
- No debe haber caracteres repetidos en el substring.
- Si hay más de un substring que satisfaga las dos condiciones anteriores, imprima el substring que aparece primero.
- Si no hay ningún substring que satisfaga todas las condiciones antes mencionadas, imprima -1.

Ejemplo:

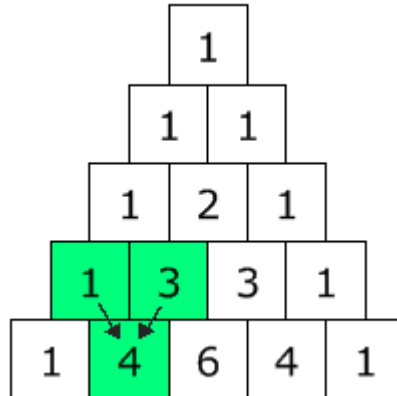
String: "caracter"

Resultado esperado: "racte"

### Problema 6:

Escribe una función de Python que imprima las primeras n filas del triángulo de Pascal.

Nota: El triángulo de Pascal es una figura aritmética y geométrica imaginada por primera vez por Blaise Pascal.



### Problema 7:

A un hombre se le dieron instrucciones para ir del punto A al punto B. Las instrucciones fueron: "SUR", "NORTE", "OESTE", "ESTE". Claramente, "NORTE" y "SUR" son opuestos, "OESTE" y "ESTE" también. Ir en una dirección y regresar en la dirección opuesta es una pérdida de tiempo y energía. Entonces, necesitamos ayudar al hombre escribiendo una función que aceptará la lista de pasos, eliminará los pasos inútiles y devolverá solo las instrucciones necesarias.

Por ejemplo: Las direcciones ["NORTE", "SUR", "SUR", "ESTE", "OESTE", "NORTE", "OESTE"] deben reducirse a ["OESTE"]. Esto se debe a que ir al "NORTE" y luego **inmediatamente** al "SUR" significa regresar al mismo lugar. Entonces los cancelamos y tenemos ["SUR", "ESTE", "OESTE", "NORTE", "OESTE"]. A continuación, vamos "SUR", tomamos "ESTE" y luego tomamos **inmediatamente** "OESTE", lo que nuevamente significa volver al mismo punto. Por lo tanto, cancelamos "ESTE" y "OESTE" para darnos ["SUR", "NORTE", "OESTE"]. Está claro que "SUR" y "NORTE" son opuestos, por lo tanto, cancelados y finalmente nos quedamos con ["OESTE"].

### Problema 8:

Desarrollar un módulo que contenga las siguientes funciones:

- Que genere 20 números enteros aleatorios entre 0 y 100 y devuelva una lista.
- Mostrar la lista obtenida por pantalla.
- Ordenar los valores de la lista y mostrarla por pantalla.

Luego crea un script main.py en el mismo directorio en el que deberás importar el módulo y ejecutar las funciones.

Nota: utilizar el módulo "random" para generar un numero aleatorio.

**Problema 9:**

Realizar el juego de “adivina el numero”. Se deberá crear un módulo que genere un valor aleatorio entre 1 y 100 y no se muestre. Después solicitar la carga de valores por teclado de un número y mostrar el mensaje “Has ganado” si el número es correcto. En caso contrario mostrar un mensaje informando si el número aleatorio es superior o inferior al introducido y solicitar un nuevo número hasta que se logre adivinar.

Luego crea un script main.py en el mismo directorio en el que deberás importar el módulo y ejecutar las funciones.

Nota: utilizar el módulo “random” para generar un numero aleatorio.

**Problema 10:**

Crea el siguiente módulo:

El módulo se denominará operaciones.py y contendrá 4 funciones para realizar una suma, una resta, un producto y una división entre dos números. Todas ellas devolverán el resultado.

En las funciones del módulo deberá de haber tratamiento e invocación manual de errores para evitar que se quede bloqueada una funcionalidad, eso incluye:

- En caso de que se envíen valores a las funciones que no sean números, deberá aparecer un mensaje que informe Error: Tipo de dato no válido.
- En caso de realizar una división por cero, deberá aparecer un mensaje que informe Error: No es posible dividir entre cero.

Una vez creado el módulo, crea un script calculos.py en el mismo directorio en el que deberás importar el módulo y ejecutar las funciones.