

Ejercicio 1

Desarrollar un algoritmo que imprima de manera ascendente los valores (todos del mismo tipo) de un diccionario.

```
# Ejemplo de uso
mi_diccionario = {"a": 3, "b": 1, "c": 2}
imprimir_valores_ascendentes(mi_diccionario)
```

Solución

```
PS C:\prueba\calculadoraPao\calculadoraPao>
[1, 2, 3]
```

Ejercicio 2

Desarrollar un algoritmo que verifique si todas las claves de un diccionario se encuentran en otro diccionario.

```
# Ejemplo de uso
diccionario1 = {"a": 1, "b": 2}
diccionario2 = {"a": 10, "b": 20, "c": 30}
print(verificar_claves(diccionario1, diccionario2))
```

Solución

```
PS C:\prueba\calculadoraPao
True
```

Ejercicio 3

Desarrollar una función que reciba dos diccionarios como parámetros y los mezcle, es decir, que se construya un nuevo diccionario con las llaves de los dos diccionarios; si hay una clave repetida en los diccionarios, se debe asignar el valor de la clave en el primer diccionario

```
# Ejemplo de uso
diccionario1 = {"a": 1, "b": 2}
diccionario2 = {"b": 3, "c": 4}
print(mezclar_diccionarios(diccionario1, diccionario2))
```

Solución

```
PS C:\prueba\calculadoraPao
{'b': 2, 'c': 4, 'a': 1}
```

Ejercicio 4

Desarrollar un programa que, dada una lista de personas, cada persona representada como un diccionario, imprima los nombres de las personas que tienen más de 100 años.

```
# Ejemplo de uso
personas = [
    {"nombres": "Pedro Julio", "apellidos": "Castro", "edad": 101},
    {"nombres": "Juan Carlos", "apellidos": "Pérez", "edad": 50},
    {"nombres": "Ana María", "apellidos": "García", "edad": 102}
]
```

```
PS C:\prueba\calculadoraPao\
Pedro Julio
Ana María
```