

# Módulo 2 - Capítulo 6

---

## Unidad 30

### Quiz 1

En el problema del intercambio de monedas, supongamos que hay una moneda de 400 wones.

Si es así, escriba el resultado de cómo el algoritmo `coin_change()` determinará el cambio de la moneda de 800 wones.

```
def coin_change(coins, amount):  
    changes = []  
    largest = 0  
  
    while(amount > 0):  
        if amount < coins[largest]:  
            largest += 1  
        else:  
            changes.append(coins[largest])  
            amount -= coins[largest]  
  
    return changes
```

```
coins = [500, 400, 100, 50, 10]  
amount = int(input('Ingrese la cantidad:'))  
  
changes = coin_change(coins, amount)  
print('Cantidad:', amount)  
print(changes, len(changes))
```

```
Cantidad: 800  
[500, 100, 100, 100] 4
```

Tomando como base el algoritmo redactado en la Unidad 30, si se tiene una moneda con una denominación de 400 wones, el código retornará 4 monedas: 1 de 500 y 3 de 100.

### Quiz 2

Suponga que el número de monedas es en problema no es infinito. Por ejemplo, si tiene las siguiente monedas en su billetera, ¿Cómo debe distribuir 710 won como cambio?

Billetera:

```
{
    500: 1,
    100: 1,
    50: 3,
    10: 2
}
```

Cambio:

```
{
    500: 1,
    100: 1,
    50: 2,
    10: 1
}
```

Lo primero es comenzar con la moneda de mayor valor, en este caso 500 won. Se debe comparar si la cantidad a cambiar es menor que el valor de la moneda, en cuyo caso se pasa a la moneda de la siguiente denominación. Si la cantidad a cambiar es mayor o igual a la moneda actual, se resta la moneda de la billetera y se aumenta la cantidad de monedas en el cambio. Por último se debe repetir el proceso anterior hasta que el cambio sea de 0 o ya no haya monedas en la billetera.

Como resultado en la billetera solo deben quedar las siguientes monedas:

```
{
    500: 0,
    100: 0,
    50: 1,
    10: 1
}
```

## Quiz 3

Dada la cantidad de monedas y la cantidad de cambio, encuentre la cantidad mínima de monedas que puede devolver como cambio.

```
def coin_change(change, coins):
    result = {}
    coins_keys = sorted(coins.keys(), reverse=True)

    for coin in coins_keys:
        coins_counter = min(change // coin, coins[coin])
        change -= coins_counter * coin
        coins[coin] -= coins_counter
```

```
        if coins_counter:
            result[coin] = coins_counter

    return result

coins = {500: 1, 100: 1, 50: 3, 10: 2}
change_amount = 710

print(coin_change(change_amount, coins))
```

```
{500: 1, 100: 1, 50: 2, 10: 1}
```