

- 1) Dada una lista de elementos (array), crear una función que retorne una nueva lista con solo los elementos que aparecen una cantidad pares de veces.

Ej:

```
["A","B","A","C","C","C","C"] -> ["A","C"]
[1,2,3,1,2] -> [1,2]
```

- 2) Crear una función que dado un string nos retorne **true** si es un isograma y **false** si no lo es. Un isograma es una palabra que no tiene letras repetidas. Consideraciones Adicionales:
 - a) Un string vacío es un isograma.
 - b) La función tiene que ser case insensitive e ignorar acentos.
 - c) Si el string tiene más de una palabra retornar false.
 - d) Se tiene que hacer clean up del string antes de comparar.
- 3) Hacer una función que reciba 2 strings (siempre los va a recibir), y devuelva si el primero termina con el segundo.

Ej:

```
func("abc", "bc"); -> true
func("abc", "d"); -> false
```

- 4) Escribir una función que dado un array de valores y retorne el mismo array pero moviendo los ceros al final.

Ej:

```
moverCeros([false, 1, 0, 1, 2, 0, 1, 3, "a"]); -> [false,1,1,2,1,3,"a",0,0]
```

- 5) Escribir una función que dado un string, retorne **true** si puede ser reordenado para formar un palíndromo o **false** si no puede ser un palíndromo.

Ej:

```
puedeSerPalindromo("ejemm"); -> true
puedeSerPalindromo("ccaabb"); -> true
puedeSerPalindromo("ccabbd"); -> false
```

- 6) Escribe una función que tome como parámetro un string con paréntesis y determina si el orden de los paréntesis es válido. La función debería retornar true si es válido o false si no lo es.

Ej:

```
"()" => true
")((()))" => false
"(" => false
"((()))((()))()" => true
```

- 7) Dado un array de números enteros, pares o impares, de al menos 3 elementos, donde todos los elementos son pares o todos los elementos son impares, excepto uno. Escribir una función que tome como parámetro el array y devuelva el elemento

aislado.

```
[2, 4, 0, 100, 4, 11, 2602, 36] => 11  
[160, 3, 1719, 19, 11, 13, -21] => 160
```