

## Auxiliar 8 - Arreglos y Ciclos

Profesor: Patricio Inostroza  
Auxiliares: Miguel Sepúlveda  
Cristóbal Loyola

10 de octubre de 2019

### P1. Recordatorio de arreglos y ciclos

Dada un arreglo de números enteros positivos, cree las siguientes funciones:

- a) La función `soloPares(L)` , que dado un arreglo de números, entregue un nuevo arreglo solo con los números pares.
- b) La función `incrementar(L,k)`, que dado un arreglo de números, modifique el arreglo, sumándole  $k$  a todos los números en ella.
- c) La función `invertir(L)` , que dado un arreglo de números, entregue un nuevo arreglo, con los elementos en orden inverso a  $L$ .

### P2. Números Primos

Implemente una función llamada `primosMenores` que, dado un número retorna el arreglo de todos los números primos menores o iguales. **Indicación:** Use la criba de Eratóstenes.

**La criba de Eratóstenes:** Es un algoritmo que permite hallar todos los números primos menores que un número natural  $N$ . Se hace un arreglo con todos los números naturales entre 2 y  $N$ . Luego se van tachando los números que no son primos de la siguiente manera:

- a) Se comienza por el 2 y se tachan todos sus múltiplos.
- b) Se avanza al siguiente número del arreglo y, si este no ha sido tachado, se tachan sus múltiplos.
- c) Se termina cuando se revisa el número `sqrt(N)`

### P3. Más listas

- a) Cree la función `quitar(L, a, b)` , que recibe como parámetros una lista  $L$  y dos números enteros  $a$  y  $b$ , que indican dos índices de la lista de python  $L$  ( $a \leq b$ ). La función debe modificar la lista, eliminando todos los elementos ubicados entre los índices  $a$  y  $b$  (inclusive) La función entrega como resultado, la cantidad de elementos que fueron borrados de la lista. Si alguno de los índices se encuentra fuera del rango de la lista, la función solo elimina los valores que estén dentro del rango. Por ejemplo:

```
>> L = [79, 45, 18, 64, 17, 24, 51]
>> quitar(L, 7,9)
```

```
>> 0
>> L
>> [79, 45, 18, 64, 17, 24, 51]
>> quitar(3,5)
>> 3
>> L
>> [79, 45, 18, 51]
```

**Indicación:** Como al usar slicing en una lista, usted puede modificar su contenido, al asignar a `L[:]` otra lista, efectivamente modifica `L` para que tenga el mismo contenido.

- b) Cree la función `shift(L,n)`, que dada una lista de elementos cualquiera y un entero `n`, entrega una nueva lista en que los elementos fueron desplazados circularmente `n` espacios a la derecha si `n` es positivo, o `n` hacia la izquierda si `n` es negativo, sin mutar `L`. Ejemplos:

```
>> L = [79, 45, 18, 64, 17, 24, 51]
>> shift(L, 2)
>> [24, 51, 79, 45, 18, 64, 17]
>> shift(L, 0)
>> [79, 45, 18, 64, 17, 24, 51]
>> shift(L, -4)
>> [17, 24, 51, 79, 45, 18, 64]
```

**Indicación:** Usted cuenta con los métodos `L.pop(i)`, y `L.insert(i, x)`, que eliminan e insertan en la lista `L`, respectivamente.