

## EJERCICIOS sobre vectores y matrices.

1. Escriba una función `vector_invert(v)` que reciba un vector de tamaño  $n$ , conteniendo valores numéricos, y obtenga el vector inverso de  $v$ . El acceso a las posiciones será por medio de índices. No usar `reverse`.

```
>>> vector_invert([1, 2, 3])  
[3, 2, 1]
```

2. Hacer una función `prod_escalar(e, v)` que reciba un escalar y un vector  $e$  implemente el producto escalar, calculando el resultado en el mismo vector de entrada.

```
>>> prod_escalar(4, [1, 3, 5, 7, 9, 1, 6])  
[4, 12, 20, 28, 36, 4, 24]
```

3. Hacer una función `prod_vector(v, w)` que reciba dos vectores del mismo tamaño  $e$  implemente el producto de vectores, que se define:

$$\sum_{i=1}^n V[i] * W[i]$$

```
>>> prod_vector([1, 3, 5], [7, 9, 1])  
39
```

4. Escriba una función `tienenmas_par(vector)` que recibe un vector de números y retorna un valor boolean que indica si el vector dado tiene más números pares que impares. La función debe mostrar un comportamiento similar a los siguientes ejemplos:

```
>>> tienenmas_par([23, 78, 1, 4, 93, 4])  
False  
>>> tienenmas_par([13, 57, 2, 4, 6])  
True
```

5. Escriba una función `inn(vector1, vector2)` que determine si todos los elementos del primer vector están contenidos en el otro vector.

```
>>> inn([1, 2, 3], [1, 2, 4, 3, 5, 6, 9, 0])
True
>>> inn([1, 2, 3, 5], [1, 2, 4, 5, 6, 7, 8, 9])
False
```

6. Escriba una función booleana recursiva `ordenado(vector)` que reciba una lista e indique si está ordenado o no.

```
>>> ordenado([1, 3, 5, 7, 9, 1, 6 ])
False
>>> ordenado([1, 3, 5, 7, 9, 11, 16 ])
True
```

7. Hacer una función `muestreM(matriz)` que reciba una matriz  $n \times m$  y la muestre en consola (usando `print`).

```
>>> muestreM([1, 2, 3], [4, 5, 6], [7, 8, 9])
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

8. Escriba una función llamada `promedio(mat)` que recibe una matriz de tamaño  $n \times m$ , y obtenga el promedio de los números en la matriz.

```
>>> promedio([[2, 1, 2, 4], [9, 8, 0, 0], [5, 0, 3, 2]])
3
```

9. Se va a recibir como entrada una matriz de tamaño  $n \times n$  o cuadrada. Debe obtenerse en forma la diagonal de la matriz de entrada, en un vector de tamaño  $n$ .

```
>>> diagonal([1, 2, 3], [4, 5, 6], [7, 8, 9])
[1, 5, 9]
```