Instituto Tecnológico de Costa Rica

Escuela de Computación

Introducción a la Programación / Taller de programación.

Profesor: Jeff Schmidt Peralta I Semestre 2018

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.

```
>>> 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 `subvector(vec1, vec2)` que reciba como entrada 2 arreglos (con acceso a sus posiciones por medio de índices) y determine si el primer arreglo es un sub-arreglo del segundo. Ejemplos: si los vectores de entrada son `[2, 3]` y `[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]` el resultado sería True. Si los vectores son: `[1, 2, 3, 5]` y `[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]` el resultado sería False.

5. Hacer una función `prod_escalarM(e, M)` que reciba un escalar y una matriz nxm e implemente el producto escalar, calculando el resultado en la misma matriz de entrada.

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

6. Hacer una función muestreM(matriz) que reciba una matriz nxm y la muestre en consola (usando print).

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

7. Se va a recibir como entrada una matriz de tamaño nxn o cuadrada. Debe obtenerse en forma la diagonal de la matriz de entrada, en un vectores de tamaño n.

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

8. Escriba una función llamada `promedio(mat)` que recibe una matriz de tamaño nxm, 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. Escriba una función **recursiva** llamada `mayor_xcol(mat)` que recibe una matriz de tamaño `nxm`, devuelva el elemento mayores de la matriz, haciendo el recorrido por columnas de la matriz original.

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