

Matrices

TEMA 1

Dada la siguiente matriz **M** y listas:

		1998		...	2003		...	2019	
		Cantidad Turistas	Capacidad Hotelera	...	Cantidad Turistas	Capacidad Hotelera	...	Cantidad Turistas	Capacidad Hotelera
Costa	Guayaquil				
		
	Manta				
Sierra
	Quito				

Oriente	Cuenca				

	Macas				
		
	El Puyo				
		

costa = ['Guayaquil', ..., 'Manta',...]

sierra = ['Quito', ..., 'Cuenca', ...]

oriente = ['Macas', ..., 'El Puyo', ...]

años = [1998, ..., 2003, ..., 2017, ..., 2019]

Escriba el código de Python para calcular las siguientes estadísticas:

1. Cantidad total de turistas que ingresaron a las ciudades del país en el año 2017. **(int)**
2. La capacidad hotelera promedio para cada una de las ciudades de la sierra considerando todos los años. **(vector de float)**
3. El número de años en los que la cantidad de turistas fue mayor que la capacidad hotelera para la ciudad de Machala. **(int)**
4. Los años en la que la capacidad hotelera del país fue superior a 5000. **(vector o lista de int)**
5. El nombre de las tres ciudades del país que más turistas en total han tenido. **(vector o lista de str)**

TEMA 2

Asuma que tiene una matriz **M** de m filas y n columnas, con valores entre 0 (totalmente sano) y 100 (totalmente enfermo) que representan el nivel de infección del Coronavirus en todos los sectores de una ciudad. Ejemplo:

0	23	37	58
0	100	94	18
21	56	42	17
11	3	68	77

Implemente lo siguiente:

1. **[10 puntos]** La función **probabilidad_contagio(M, f, c)** que recibe la matriz de niveles, los índices de fila y columna de una celda de la matriz y retorna la probabilidad de contagio para esa ubicación. La probabilidad se calcula como el promedio de los niveles de infección de **todas las celdas vecinas a f,c. La celda (f,c) nunca estará en el borde de la matriz.**

Asuma que ya tiene implementada la función **vacunar(M, f, c, dosis)** que actualiza la matriz M disminuyendo los niveles de infección en varias de sus celdas.

2. **[20 puntos]** Un programa principal que repita los siguientes pasos hasta que las probabilidades de contagio para las celdas (1,1) y (1, n - 1) de la matriz **M** sean menor a 20 cada uno:
 - A) Obtenga, de la matriz de niveles, los índices de fila y columna de la celda más infectada
 - B) Genere aleatoriamente un **valor entero par** de dosis entre 2 y 200
 - C) Usando la función **vacunar**, aplique la dosis del paso B) en la celda del paso A)

TEMA 3

Asuma que ya existe implementada una función **votar** que recibe una lista de películas y retorna una matriz **M** de **posiciones**, en el siguiente formato:

	Parásitos	Mujercitas	Joker	...	1917
Juez 1	1	4	2	...	3
Juez 2	1	3	2	...	4
Juez 3	2	5	1	...	8
...
Juez n	3	7	2	...	4

Las celdas representan las posiciones en que cada juez ha ubicado a cada una de las películas según su criterio, **siendo uno (1) la posición para la mejor película**. Por ejemplo, el Juez 3 ubicó **Joker** como la **mejor película**, **Parásitos** en segunda posición y **Mujercitas** en quinta posición. Asuma que también tiene las siguientes listas:

```
jueces = ['Juez 1', 'Juez 2', 'Juez 3', ..., 'Juez n']  
peliculas = ['Parasitos', 'Mujercitas', 'Joker', ..., '1917']
```

Implemente lo siguiente:

1. **[20 puntos]** La función **analizar_matriz(M)** que recibe la matriz de posiciones **M**. La función retorna una tupla con el índice de la película que fue considerada como **mejor película** por la mayoría de los jueces, el porcentaje de jueces que la escogieron y el índice de la película que fue considerada **mejor película** por la menor cantidad de jueces.
2. **[10 puntos]** La función **jueces_película(M, pelicula, peliculas, jueces)** que recibe la matriz de posiciones **M**, el nombre de una película, la lista de películas y la lista de jueces. La función retorna un vector con los nombres de todos los jueces que votaron por esa película como **mejor película**.
3. **[15 puntos]** Un programa principal que muestre por pantalla el nombre de la **mejor película y los jueces que votaron por ella**. La mejor película se determina usando el siguiente algoritmo:
 - a. Los jueces votan por las películas candidatas usando la función **votar** que recibe la lista de películas elegibles.
 - b. Verificar que la opción más votada como **mejor película tenga más del 50% de los votos**.
 - i. En caso de que esto suceda, esa película se determina como la ganadora.

En caso contrario, la película menos votada como mejor película se elimina de la lista de elegibles y se **repite el proceso desde el paso a paso**.

TEMA 3

La World Wide Fund for Nature (WWF) se encuentra realizando el #TenYearChallenge del Ártico. Para esto tiene información sobre la cantidad de hielo y población de diversas especies durante los años 2009 y 2019 en un archivo con el siguiente formato: Número de filas

Número de columnas

Año, fila, columna, hielo, especie

Ejemplo:

22

14

Año, fila, columna, hielo (0/1), especie ...

2009, 14, 8, 1, 9 ...

2019, 5, 7, 0, 3

Implemente las siguientes funciones:

1. **[15 puntos] crearMatriz(nomArchivo)** que recibe el nombre del archivo con los datos de la WWF y retorna una tupla que contiene las siguientes cuatro matrices:

Cantidad de hielo

2009

2019

Especies de animales

2009

2019

1	0	0	1	...
0	1	1	1	...
1	0	1	0	...
1	1	1	0	...
0	1	0	1	...
1	1	1	0	...
...

1	0	0	1	...
0	1	0	1	...
1	0	1	0	...
1	0	1	0	...
0	1	0	1	...
0	0	1	0	...
...

1	2	4	4	...
2	5	5	3	...
1	3	9	1	...
1	1	4	2	...
4	22	4	7	...
1	1	14	4	...
...

3	2	4	1	...
0	11	3	5	...
1	0	67	1	...
2	22	3	2	...
13	13	0	2	...
3	0	2	3	...
...

En las matrices de hielo cada valor uno (1) en una celda representa la presencia de hielo. En las matrices de especies cada celda representa la presencia de un animal de una especie dependiendo de una cierta codificación, ejemplo:

0: No hay animal 1: Lobo ártico 2: Oso Polar Para 3: Reno 4: Foca 5: ...

el resto del examen considere lo siguiente:

Las matrices pueden ser divididas en cuadrantes (**cuatro subregiones de igual tamaño**) llamados Q1, Q2, Q3 y Q4, como se muestra abajo:

	Q1		Q2
	Q3		Q4

Asuma que existen las siguientes funciones:

- cuadrantes(matriz)** que recibe una matriz y devuelve una tupla con los cuadrantes de la matriz, es decir, retorna (Q1, Q2, Q3, Q4) que son a su vez matrices de Numpy.
- poblacionEspecie(mAnimales, especie)** que recibe una matriz de animales y el código de una especie. La función retorna una tupla con la cantidad total de animales de esa especie para cada uno de los cuadrantes. La tupla tendrá el siguiente formato (pQ1, pQ2, pQ3, pQ4).

2. **[10 puntos] densidadHielo(mHielo)** que recibe una matriz de hielo y retorna una tupla con la densidad de hielo para cada cuadrante. La densidad de un cuadrante es la cantidad total de sus celdas con valor uno (1) dividido para el número total de celdas del cuadrante. La tupla tendrá el siguiente formato (dQ1, dQ2, dQ3, dQ4)
3. **[20 puntos] especieDominante(mAnimales)** que recibe una matriz de animales y retorna una tupla con los códigos de la especie que más se repite (dominante) en cada cuadrante. La tupla tendrá el siguiente formato (eQ1, eQ2, eQ3, eQ4).
4. **[15 puntos] migracionEspecie(mAnimales2009, mAnimales2019, especie)** que recibe la matriz de animales del 2009, la matriz de animales del 2019 y el código de una especie. La función debe retornar una tupla con dos strings. El primer valor corresponde al cuadrante (' Q1 ' , ' Q2 ' , ' Q3 ' o ' Q4 ') donde hubo mayor población de animales de la especie en el 2009. El segundo valor corresponde al cuadrante donde hay mayor población de animales de la especie en el 2019.

Para el siguiente numeral, asuma que tiene un diccionario con los nombres de **todas** las especies del ártico en el siguiente formato:

```
dicEspecies = {codigo_especie:nombre_especie}
```