Guía 02 de Python

Programación

Ejercicio 1 - Particionando listas

Implementar una función partList que tome una lista y un número entero k como entrada, y devuelva una lista donde todos los elementos menores que k aparezcan antes de los elementos mayores o iguales que k, manteniendo el orden relativo de los elementos.

Ejemplo:

```
partList([4, 3, 2, 1, 5, 6] ,4) => [3, 2, 1, 4, 5, 6]
```

Ejercicio 2 -Rotando listas

Implementar una función rotateList que tome una lista y un número entero k como entrada, y rote la lista k veces hacia la derecha.

Ejemplo:

```
rotateList([1, 2, 3, 4, 5],2) => [4, 5, 1, 2, 3]
```

Ejercicio 3 - Sumando matrices

Escribe una función sumMatrix que tome dos matrices como entrada y devuelva una nueva matriz que sea el resultado de la suma de las dos matrices de entrada. Las matrices deben tener la misma dimensión (mismo número de filas y columnas) en caso que no lo cumplan se debe avisar al usuario a través de la consola del sistema.

Ejemplo:

```
SumMatrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]],[[9, 8, 7], [6, 5, 4], [3, 2, 1]])
=> [[10, 10, 10], [10, 10, 10], [10, 10, 10]]
```

Ejercicio 4 - Multiplicando matrices

Escribe una función multMatrix que tome dos matrices como entrada y devuelva una nueva matriz que sea el resultado de la multiplicación de las dos matrices de

entrada. Las matrices deben cumplir la siguiente condición: El número de columnas de la primera matriz debe ser igual al número de filas de la segunda matriz en caso que no lo cumplan se debe avisar al usuario a través de la consola del sistema.

Ejemplo:

MultMatrix([[1, 2], [3, 4], [5, 6]],[[7, 8, 9], [10, 11, 12]]) => [[27, 30, 33], [61, 68, 75], [95, 106, 117]]

Ejercicio 5 - Producto escalar

Escribe una función Scalar que tome dos matrices como entrada y devuelva su producto escalar.

Ejemplo:

Scalar([[1, 2], [3, 4]],[[2, 0], [1, 2]]) => [[4, 4], [10, 8]]

Ejercicio 6 - Invertir diccionario

Implementar una función llamada invert que tome un diccionario como entrada y devuelva un nuevo diccionario donde las claves sean los valores originales y los valores sean listas de las claves originales que tenían ese valor.

Ejemplo:

Scalar({'a': 1, 'b': 2, 'c': 1, 'd': 3}) => {1: ['a', 'c'], 2: ['b'], 3: ['d']}

Ejercicio 7 - Promedios de claves

Implementar una función llamada averageKeys que tome un diccionario donde los valores son listas de números y devuelva un nuevo diccionario donde los valores sean los promedios de las listas correspondientes.

Ejemplo:

averageKeys({'a': [1, 2, 3], 'b': [4, 5, 6], 'c': [7, 8, 9]}) => {'a': 2.0, 'b':5.0, 'c': 8.0}

Ejercicio 8 - Fusionando diccionarios

Implementa una función mixDict que tome dos diccionarios como entrada y devuelva un nuevo diccionario que contenga todas las claves de ambos diccionarios y sus valores correspondientes. Si hay claves duplicadas, suma los valores.

Ejemplo:
mixDict({'a':1, 'b':2, 'c':3},{'b':3, 'c':4, 'd':5})=>{'a':1, 'b':5, 'c': 7, 'd': 5}