Ejercicios/_1ejercicios.py

```python
# Se define una lista de tuplas con información sobre los usuarios y sus amigos en una red social
red_social = [("Juan", ["Maria", "Pedro", "Luis"]), ("Maria", ["Juan", "Pedro", "Juan"]), ("Pedro", ["Juan", "Maria"]), ("Luis", ["Juan"])]

# Se crea una nueva lista sin duplicados, eliminando las cuentas repetidas de amigos
#for i in range(len(red_social)):
    #usuario = red_social[i][0]
    #amigos = red_social[i][1]
#for tupla in red_social:
red_social_sin_duplicados = []
for usuario, amigos in red_social:
    # Se crea una lista de amigos sin duplicados
    amigos_sin_duplicados = list(set(amigos))
    # Se agrega una nueva tupla a la lista red_social_sin_duplicados con el usuario y la lista de amigos sin duplicados
    red_social_sin_duplicados.append((usuario, amigos_sin_duplicados))

# Se crea una nueva lista que almacena el número de amigos de cada usuario
amigos_por_usuario = []
for usuario, amigos in red_social_sin_duplicados:
    # Se agrega el número de amigos de cada usuario a la lista amigos_por_usuario
    amigos_por_usuario.append((usuario, len(amigos)))

# Se convierte la lista amigos_por_usuario a una tupla para hacerla inmutable
amigos_por_usuario = tuple(amigos_por_usuario)

# Se imprime la lista completa de usuarios y su número de amigos correspondiente
print("Usuarios con número de amistades:", amigos_por_usuario)

# Se obtiene el usuario con más amigos, extrayendo el índice del máximo en la lista numero_amigos
lista_usuarios = [tupla[0] for tupla in amigos_por_usuario]
numero_amigos = [tupla[1] for tupla in amigos_por_usuario]

indice_con_mas_amigos = numero_amigos.index(max(numero_amigos))
usuario_con_mas_amigos = lista_usuarios[indice_con_mas_amigos]

# Se imprime el usuario con más amigos
print("Usuario con mayor conexión:", usuario_con_mas_amigos)

# Output: tuplas de tuplas -- usuario, número de amigos
# Output: Usuario con más amigos


"""set Clientes"""

# Se definen dos bases de datos como listas de tuplas con información sobre clientes
base_datos1 = [("Juan", "juan@example.com", "555-1234"), ("Maria", "maria@example.com", "555-5678"), ("Pedro", "pedro@example.com", "555-9012")]
base_datos2 = [("Juan", "Calle 123", ["Libro1", "Libro2"]), ("Maria", "Calle 456", ["Libro3"]), ("Luis", "Calle 789", ["Libro4"])]
```

```
47
48  # Se crean dos conjuntos de nombres de clientes en ambas bases de datos, utilizando una
    comprensión de listas
49  nombres1 = set([cliente[0] for cliente in base_datos1])
50  nombres2 = set([cliente[0] for cliente in base_datos2])
51
52  # Se encuentra la intersección de ambos conjuntos, es decir, los clientes que aparecen en
    ambas bases de datos
53  nombres_comunes = nombres1.intersection(nombres2)
54
55  # Se imprime la lista de nombres de clientes comunes
56  print(nombres_comunes)
57
58  # Se crea una lista de tuplas de clientes comunes, recorriendo las dos bases de datos
    mediante un bucle for anidado
59  # y comprobando si el nombre del cliente aparece en la lista de clientes comunes.
60  clientes_comunes = []
61  for cliente1 in base_datos1:
62      if cliente1[0] in nombres_comunes:
63          for cliente2 in base_datos2:
64              if cliente2[0] == cliente1[0]:
65                  # Si el cliente coincide en ambas bases de datos, se agrega una nueva tupla
    a la lista clientes_comunes
66                  # con la información del cliente de ambas bases de datos
67                  clientes_comunes.append((cliente1[0], cliente1[1],cliente1[2], cliente2[1],
    cliente2[2]))
68                  break
69
70  # Se imprime la lista completa de clientes comunes
71  print(clientes_comunes)
72
73
74
75
76  """Lista de tuplas"""
77  # Se define una lista de tuplas con información sobre los libros y sus autores
78  lista_libros = [('El aleph', 'Jorge Luis Borges'), ('Cien años de soledad', 'Garbriel Garcia
    Márquez'), ('La ciudad y los perros', 'Mario Vargas Llosa')]
79
80  # Se crea una lista vacía para almacenar los títulos de los libros y los apellidos de los
    autores
81  titulos_y_apellidos = []
82
83  # Se recorre la lista de libros con un bucle for, desempaquetando cada tupla en título y
    autor
84  for tupla in lista_libros:
85      titulo, autor = tupla
86
87      # Se utiliza el método split() para dividir el nombre completo del autor en una lista de
    palabras,
88      # y se selecciona el último elemento de la lista (que debería ser el apellido)
89      apellido = autor.split()[-1]
90
91      # Se agrega una nueva tupla a la lista titulos_y_apellidos, que contiene el título del
    libro
```

```
    # y el apellido del autor
    titulos_y_apellidos.append((titulo, apellido))

# Se imprime la lista completa de títulos de libros y apellidos de autores
print(titulos_y_apellidos)



for num in range (2,101):
    primo=True
    for i in range (2,num):      # while primo == True and i < num:
        if num%i == 0:
            primo=False
            break                # i+=1
    if primo:
        print(num)


```