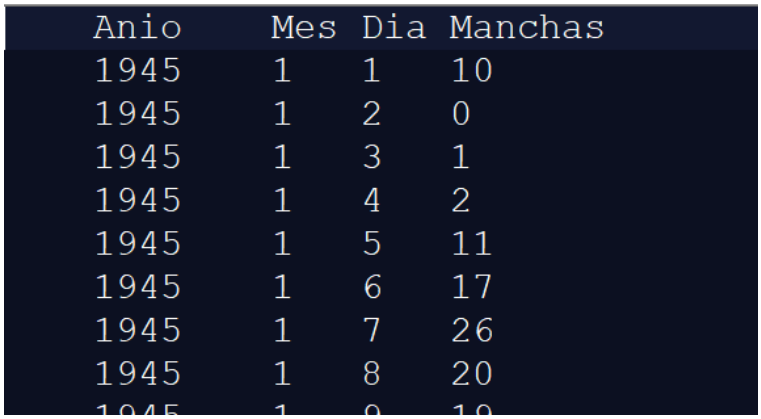
Fundamentos de Programación

Tarea Archivos/Colecciones

Ejercicio 1

Como astrónomo, su institución ha capturado la cantidad de manchas solares (https://es.wikipedia.org/wiki/Mancha_solar) observadas cada de día (archivos **manchas.csv**) desde el 1 de Enero de 1945 hasta el 31 de Diciembre de 2016. Le dan esa información en formato csv, donde los datos están separados por un tab (\t):



Anio	Mes	Dia	Manchas
1945	1	1	10
1945	1	2	0
1945	1	3	1
1945	1	4	2
1945	1	5	11
1945	1	6	17
1945	1	7	26
1945	1	8	20

Ud. va escribir un programa para analizar esta información. Para esto, deberá cargar la información en un diccionario con la siguiente estructura:

```
datos_manchas = {
      (1,1,1945):  10,
      (2, 1, 1945): 0,
      …
      (31,12,2016): 6
}
```

Entonces, deberas cargar el archivo **al iniciar el programa solamente**, y luego implementa la siguiente funcionalidad (mostrando un menú):

1. Una opción, que pida al usuario un numero de manchas, y muestre todas las fecha donde hubieron esa cantidad de manchas o más. Al final, mostrar el promedio de manchas en los días seleccionados.
2. Una opción que pida una fecha como letra (por ejemplo 29 octubre 2021), y muestre la cantidad de manchas en ese día. **TIP: crea una función que reciba la fecha como texto (con el formato mostrado arriba), y que te devuelva la fecha como tupla de números (para que la puedas usar directamente en el diccionario).**

3. Una opción que tome dos fechas, y muestra , para cada cantidad de manchas encontradas en ese rango, en cuantos días se encontraron esa cantidad de manchas. Para esto genera un diccionario, donde la clave es la cantidad de manchas, y el valor es la cantidad de días donde se encontraron esa cantidad de manchas:

```
{
     15: 20,
     10: 3,
     …
}
```

Por ejemplo, en el diccionario de arriba, hubo 20 días del año donde en cada uno de ellos se observó 15 manchas solares.

Para saber si una fecha es mayor o igual que otra, usa estas condiciones**.**
TIP: crear una función que reciba dos tuplas de fecha y aplique las condiciones:

1. Si el mes de la fecha de inicio es igual al mes de la fecha de fin:
    a. Si el año es igual, el día de la fecha de fin debe ser mayor o igual al día de inicio
    b. Si los años no son iguales, el año de fin debe ser mayor o igual al de inicio
    c. Caso contrario la fecha de fin es anterior a la de inicio
2. Si el mes de le fecha de inicio NO es igual al mes de la fecha de fin:
    a. Si el año de fin es mayor al de inicio, la fecha de fin es posterior a la de inicio
    b. Si el año de fin igual al de año de inicio, entonces el día y el mes de fin debe ser mayor o igual al de inicio
    c. Caso contrario, la fecha de fin es anterior a la de inicio

Usa funciones para escribir tu programa.

Ejercicio 2

Como administrador de una aerolínea, te dan dos archivos, con la información de aerolienas (**aerolíneas.csv**), y (**rutas.csv**). El archivo aerolíneas.csv esta organizado de la siguiente forma:

codigo_aerolinea,pais,nombre

Y el archivo de aerolíneas de la siguiente forma:

codigo_aerolinea,aeropuerto_salida,aeropuerto_destino,paradas

Las rutas de las aerolines se especifican en el formato aeropuerto_salida-aeropuerto_destino, por ejemplo:

GYE-UIO

Es la ruta Guayaquil Quito. Para llevar registro de esto, generaras dos diccionarios: el de aerolíneas, donde la clave es el código de aerolínea, y el valor un diccionario con el país y nombre de la aerolínea:

```
aerolíneas = {
     'AA': {
          'nombre': 'American Airlines',
          'pais': 'United States'
     },
     'AV': {
          'nombre': 'Avianca',
          'pais': 'Colombia'
     },
     …
}
```

Y el diccionario de rutas, donde la clave es el código de aerolínea, y el valor un conjunto de strings de rutas, en formato 'fuente-destino'.

```
rutas = {
     'AA': { 'GYE-MIA', 'JFK-MIA', …}
     'AV': { 'BOG-GYE', 'VVI-LIM', … },
     …
}
```

TIP: crea una funcion que reciba el nombre de los DOS archivos y retorne los dos diccionarios.

Luego escribiras un programa que tendrá las siguientes opciones. **USA funciones para escribir tu programa.**

1. Una opción que pida los nombres de aerolíneas (no códigos), separados por comas, y muestra las rutas que cubren esas aerolíneas. Ejemplo de salida:

```
Aerolineas: american Airlines,Avianca

American Airlnes
GYE-MIA
JFK-MIA
…

Avianca
BOG-GYE
VVI-LIM
…
```

2. Una opción, que pida una ruta, y muestre las aerolíneas que cubren esa ruta. Por ejemplo:

```
Ruta: bog-gye

Aerolineas que cubren la ruta BOG-GYE:

Avianca (AV)

LATAM (LA)
```

3. Una opción que nos permita agregar una ruta a una aerolínea. Pide el nombre de aerolínea y la ruta en formato fuente-destino

4. Una opción que nos permita eliminar una ruta. La ruta deberá ser eliminada de todas las aerolíneas donde aparezca esa ruta.

5. Una opción para agregar una aerolínea. Pide el nombre, código y país separados por comas. Si la aerolínea existe, muestra el mensaje “la aerolínea ya existe”

6. Una opción para eliminar una aerolínea. Pide el nombre de aerolínea. Tambien deberas eliminar todas su rutas del diccionario de rutas.