Deusto Facultad de Ingeniería Ingeniaritza Fakultatea	EXAMEN DE PROGRAMACIÓN I	07-FEBRERO-2023

Nombre: _____

DNI: _____ ***Aula:*** _____ ***Fila:*** _____ ***Columna:*** _____

Instrucciones

- El ejercicio a pie de máquina se entregará al terminar el examen. Se recogerán en pendrive por parte del profesor.
- NO está permitida la conexión a internet en el examen. Tampoco la utilización de tu propio portátil ni la consulta de apuntes o de otros códigos.
- Descarga el fichero llamado 'series.csv' de ALUD y guárdalo en una **carpeta** que se llame "**ExamenFinal-NombreApellidos**", por ejemplo: ExamenFinal-PabloGaraizar.
- Guarda el código de todos los ejercicios en **un único fichero** dentro de la carpeta anterior que se llame "**ExamenFinal-NombreApellidos.py**", por ejemplo: ExamenFinal-JennyFajardo.py.

Nos piden desde Netflix ayuda para poder desarrollar su sistema de recomendación. Para ello, lo primero que necesitan es analizar los datos que tienen de las notas de las series que tienen en catálogo. Además, nos piden trabajar con una versión simplificada de su modelo de datos en el que hay estas dos entidades:

Serie

Almacena los datos que nos interesan de cada serie para este estudio:

- **titulo**: título de la película (str), por ejemplo: 'Spider-man'
- **duración**: duración media del capítulo (float).
- **género**: género(s) de la película (lista de string).
- **rating**: valoración media de la serie (float).
- **votos**: número de votos que ha conseguido la serie (int).
- **inicio**: año en el que comenzó a emitirse la serie (int).
- **fin**: año en el que dejó de emitirse la serie (int). En caso de que la serie se siga emitiendo, se almacena el año 2023.

User

Almacena los datos que nos interesan de cada cuenta de Netflix en este estudio:

- **username:** cadena de caracteres (string), por ejemplo: 'bosanz'.
- **series:** lista de series (lista de elementos de la entidad descrita anteriormente).

Un ejemplo de user con varias series podría ser:

```
user1 = {
  'username': 'bosanz',
  'series': [
  {'titulo': "L.A.'s Finest",'duracion': 60.0,'generos': ['Action', 'Comedy', 'Crime'],
'rating': 6.0, 'votos': 4864,'inicio': 2019,'fin': 2020},
  {'titulo': 'Breaking Bad: Original Minisodes','duracion': 45.0, 'generos': ['Short', '
Comedy', ' Crime'],'rating': 7.0,'votos': 1207,'inicio':2009,'fin': 2011},
  {'titulo': 'Dongbaekkkot Pil Muryeop', 'duracion': 27.0, 'generos': ['Comedy', 'Drama',
'Romance'], 'rating': 8.0, 'votos': 1687, 'inicio': 2019,'fin': 2023}
]
}
```

Teniendo esto en cuenta, programa las siguientes funciones:

- **datos_prueba:** función que recibe una lista de users vacía y la rellena introduciendo al menos 2 users que tengan al menos 3 series cada uno y 2 géneros en cada serie. Define estos datos con variables locales de esta función y añade los users a la lista recibida (no es necesario solicitar información mediante input) **(0,5 puntos).**

- **cargar_series:** función que carga el fichero 'series.csv' en una estructura de datos que puede ser una lista de dos dimensiones o un diccionario y devuelve esa estructura de datos **(1 punto).** En caso de no poder cargar el fichero, crea una lista provisional similar**:**

```
lista_series = [{'titulo': 'L.A.'s Fines','duracion': 60.0, 'generos': ['Action', 'Comedy', 'Crime'], 'rating': 6.0, 'votos': 4864, 'inicio': 2019, 'fin': 2020},
 {'titulo': 'Breaking Bad: Original Minisodes', 'duracion': 45.0, 'generos': ['Short', 'Comedy', 'Crime'], 'rating': 7.0, 'votos': 1207, 'inicio': 2009, 'fin': 2011},
 {'titulo': 'Dongbaekkkot Pil Muryeop', 'duracion': 27.0, 'generos': ['Comedy', 'Drama', 'Romance'], 'rating': 8.0, 'votos': 1687, 'inicio': 2019, 'fin': 2023},…]
```

- **titulo_serie_mas_corta:** función que recibe una lista con las series y devuelve el título de la serie cuya duración de capítulos es más corta **(1 punto).**

- **serie_mas_tiempo_antena:** función que recibe la lista de series y que devuelve los datos de la serie que más tiempo lleva en antena, es decir, la que tiene una mayor diferencia entre el año inicial y el final **(1.5 puntos).**

- **crear_usuarios:** función que recibe una lista vacía de usuarios y una lista de series, la completa y devuelve una lista con 500 usuarios y con al menos 3 series seleccionadas al azar (no puede estar repetida). Los username deben de ser consecutivos, y el usuario debe concatenar el texto "usuario-" con el username **(2 puntos).**

- **usuarios_por_genero**: función que recibe la lista de usuarios completa y devuelve un diccionario en el que las claves son los distintos géneros y el valor el número de usuarios que han visto series de ese género **(2 puntos).**

- **guardar_rating:** función que recibe la lista de los usuarios y almacena en el fichero 'rating.csv' la lista de los username y nota media de todas las series que ha visualizado el usuario **(2 puntos).** Por ejemplo:

```
user-0;7.333333333333333
user-1;7.0
user-2;6.666666666666667
```

Tu programa principal debería ser algo así:

```
users = [] # Lista de users vacía

datos_pruebas(users) # Creamos los users de prueba

series = cargar_series() # Cargamos datos desde series.csv

print(titulo_serie_mas_corta(series)) # Mostramos el título de la serie más corta

print(serie_mas_tiempo_antena(series)) # Mostramos el título de la serie de más tiempo

users = crear_usuarios(users, series) # Creamos 500 usuarios con al menos 3 series

print(usuarios_por_genero(users)) # Mostramos el diccionario de usuarios por género

guardar_ratings(users) # Guardamos la nota media de cada usuario en rating.csv
```