



Taller de Programación

Examen

27 de junio de 2023

Resuelva individualmente los dos problemas descritos más abajo. Cada pregunta debe estar contenida en un único archivo en con el formato

`ex_apellido_nombre_pn.py` o en `ex_apellido_nombre_pn.ipynb`, donde n es el número de la pregunta. Estos archivos deben ser entregados en un archivo comprimido .zip con el formato `ex_apellido_nombre.zip` en canvas.

Detalles de la Entrega

- La fecha de entrega es el 7 de julio del 2023 a las 23:59 hrs.
- Si su código tiene errores de sintaxis (no se ejecuta), su tarea no se considerará entregada.
- Puede programar funciones adicionales si lo estima conveniente.
- No puede utilizar la librería `Pandas`
- Recuerde que solo puede compartir sus soluciones con su profesor(a) y los ayudantes del curso.
- Su solución, junto con las soluciones de todas las secciones, será evaluada por un software detector de plagio. De detectarse copia, su certamen será evaluado con un 1.0 y la situación será informada al comité de ética de la Facultad, lo que puede derivar en una causal de eliminación de la universidad.
- Su profesor, en caso de que dude de la autoría del código, puede interrogarlo sobre este para demostrar que no hubo plagio. En caso de que no pueda explicar su código será calificado con nota 1.0 y denunciado al comité de ética de la Facultad.

1.0 Juego el Ahorcado (3.0 pts)

El juego del ahorcado consiste en adivinar una palabra aleatoria letra por letra. Programe la versión especial del juego del “ahorcado” utilizando las funciones descritas abajo y considerando que el jugador puede ingresar hasta cinco letras incorrectas (5 intentos), y que una vez agotados los intentos el usuario puede adivinar la palabra ingresándola directamente hasta un máximo de tres intentos.

1. **Función** `palabras(lista_palabras)` [0.2 pts.]: Retorna la palabra a adivinar de una lista predefinida `lista_predefinida` de manera aleatoria.
2. **Función** `existeletra(letra, palabra)` [0.2 pts.]: Recibe una letra y una palabra y comprueba si la letra existe en la palabra, retornando `True` o `False` según sea el caso.
3. **Función** `pantalla()` [1.0 pts] Imprime letra por letra la palabra a medida que el usuario va adivinando las letras de la palabra, dejando espacios vacíos para las letras que no han sido adivinadas.
4. **Función** `ganar()` [0.3 pts.]: Comprueba si se ha completado la palabra clave, devolviendo `True` o `False` dependiendo del caso.
5. **Función** `tresintentos()` [0.6 pts.]: Permite al usuario adivinar la palabra final en tres intentos, después de haber acumulado 5 errores. Si el usuario logra adivinar la palabra esta debe imprimir en pantalla "HAS ACERTADO" y en caso contrario "NO HAS ACERTADO" o un mensaje similar.
6. **Código principal** [0.7 pts.]. Permite integrar todas las funciones anteriores y ejecutar el juego. No es necesariamente una función.

Nota 1: Algunas funciones requieren el uso de otras funciones dentro de estas.

Nota 2: Debe programar el juego descrito utilizando las funciones descritas y puede incluir otras funciones adicionales. La entrega de un juego que no incluya las funciones descritas se considerará como código **NO ENTREGADO**.

2.0 Películas IMDB (3.0 pts)

El archivo `cast.csv` contienen información sobre todas las películas en la base de películas (IMDb). Estos incluyen el título, el año de estreno, los personajes y los actores y actrices de cada una de ellas. Con estos archivos y los aprendidos en clases, resuelva los problemas descritos a continuación:

1. **Películas por actor** [0.2 pts.] Programe la función `films_actor(file, actor)` que recibe el archivo `cast.csv` y retorna una **lista** con el nombre de todas las películas en

las que actúa el actor o actriz actor.

2. **Partners [0.8. pts.]** Programe la función `partners(file, actor1, actor2)` que recibe el archivo `cast.csv` y el nombre de dos actores o actrices y retorna una **lista** de todas las películas que la que ambos han participado.
3. **Batman vs Superman [2.0 pts.]**. Definiendo un año "Batman" como el año donde el personaje Batman aparece más veces que Superman, y un año "Superman" como el año donde el personaje Superman aparece más veces que Batman, programe la función `batman_vs_superman(file)` que recibe el archivo `cast.csv` y retorna un diccionario cuyas claves corresponden a un año, y los valores contienen el string `Batman` o `Superman`, dependiendo de si es un año "Batman" o "Superman". Considere sólo los años en que ambos personajes aparecen en al menos una película.