UNIVERSIDAD ADOLFO IBÁÑEZ

Sobre la estudiatón:

¡Muchísimas gracias por tu asistencia! Queremos asegurarnos que te hayas inscrito y registrado tu ingreso con una foto al inicio o durante la actividad. Habrán ayudantes tomando la asistencia, **asegúrate que registren tu RUT y una foto**/selfie.

Si tienes más ideas para complementar el ejercicio y aumentar su dificultad, ¡adelante! No te limites y compártelas con tus compañeros.

Recomendamos igualmente intentar ir resolviendo los desafíos de forma individual y a mano cosa de ir soltándose y descubrir los errores más comunes para no repetirlos en la prueba.

Durante el evento se acercará personal del staff para llevarlos a la zona de entrega de pizza y bebida cosa de evitar aglomeraciones. En esta ocasión estamos vendiendo cada trozo de pizza a $1.000, directamente coordinado con los Centros de Alumnos, la bebida si es de libre consumo. Deberán entonces solicitar tu sticker de pizza en la zona de caja, para poder retirar el o los trozos que compraste. Tenemos por supuesto una capacidad limitada por lo que limitaremos la venta si la demanda inicial es muy alta.

En el siguiente link / QR podrán encontrar este material en versión digital además de material complementario para el estudio de la prueba 3.

Alrededor de las **18:00 habilitaremos un link para entregar feedback (en el mismo QR)** y poder mejorar este tipo de eventos. Si completaste la encuesta final y por supuesto asististe a la estudiatón, estarás **participando por entradas dobles al cine**.

Información sobre la prueba:

*La prueba 3 de este sábado 8 de Julio durará igual que la anterior 120 minutos, empezaremos a las **11:40 am**, entre ordenar el ingreso y leer la prueba a más tardar terminamos 14:10.*

*La estructura será similar a las pruebas anteriores haciendo **énfasis** en los contenidos nuevos no evaluados anteriormente, es decir **funciones y archivos**.*

Por supuesto es individual y en papel, será escaneada por lo que se recomienda uso de lápiz pasta.

***No existirá bonus** en esta oportunidad, dado que debemos corregir la prueba en tiempo récord para cerrar el semestre.*

UNIVERSIDAD ADOLFO IBÁÑEZ

Desafío 1 - Pokédex

Para poner en práctica estos conceptos, se ha decidido crear una Pokédex básica que permita buscar los datos de los primeros 150 Pokémon.

Para empezar, necesitamos crear un archivo CSV que contenga los nombres, alturas, pesos, tipos y la información de evolución de los 150 Pokémon iniciales. Cada línea del CSV debe tener el siguiente formato:

```
Nombre,Altura,Peso,Tipo,Evolución
Bulbasaur,0.7,6.9,Planta/Veneno,Inicial
Ivysaur,1.0,13.0,Planta/Veneno,Evolución de 1
Venusaur,2.0,100.0,Planta/Veneno,Evolución de 2
...
```

El campo "Tipo" indica el tipo o tipos del Pokémon (por ejemplo, Planta/Veneno). El campo "Evolución" indica si el Pokémon es inicial o una evolución, especificando el número (id) del Pokémon predecesor en caso de ser una evolución.

1.- Cree un programa que genere este archivo CSV de ejemplo con los datos de los primeros 150 Pokémon, incluyendo la información de evolución y los tipos de cada Pokémon. Asuma que cuenta con una lista que tiene la información de todos los pokémon. NO ES NECESARIO que escriba toda la lista a mano, con los primeros 2 es suficiente.

2.- Implemente una función llamada ***buscar_pokemon*** que reciba como parámetros el nombre de un Pokémon, el tipo del Pokémon o ambos, y el nombre del archivo CSV. Esta función deberá buscar los Pokémon que cumplan con los criterios de búsqueda en el archivo CSV y mostrar en pantalla sus características básicas de altura, peso, tipos y su información de evolución (si corresponde). Si no se encuentra ningún Pokémon que cumpla con los criterios de búsqueda, se deberá mostrar un mensaje indicando que no se encontraron Pokémon.

- Si se especifica solo el nombre del Pokémon, se buscará el Pokémon exacto por su nombre.
- Si se especifica sólo el tipo del Pokémon, se listarán todos los Pokémon que tengan ese tipo, incluso si el tipo especificado es uno de los tipos del Pokémon.
- Si se especifica tanto el nombre como el tipo del Pokémon, se buscará el Pokémon por nombre y tipo.

3.- Finalmente, en el programa principal, pide al usuario que ingrese los criterios de búsqueda (nombre y/o tipo) y utiliza la función ***buscar_pokemon*** para mostrar los Pokémon que cumplen con esos criterios.

UNIVERSIDAD ADOLFO IBÁÑEZ

UNIVERSIDAD ADOLFO IBÁÑEZ

Desafío 2 - Resolviendo un laberinto

En el clásico juego del Buscaminas, los jugadores seleccionan un nivel de dificultad y luego ingresan las coordenadas de las casillas que desean descubrir o marcar con una bandera.

El programa debe permitir al jugador elegir entre tres niveles de dificultad: fácil, medio y difícil. Cada nivel tiene un tamaño de tablero predefinido y un número específico de minas ocultas (a libre elección). Los jugadores ingresarán las coordenadas de las casillas dónde desean realizar una acción. Las acciones son, descubrir la casilla o colocar una bandera. Si el jugador selecciona una casilla con una mina, el juego termina. Si el jugador coloca una bandera en todas las minas y todas las demás casillas están descubiertas, el juego termina y el jugador gana la partida.

Debes implementar al menos las siguientes funciones:

- ***crear_tablero(nivel)***: Esta función recibe como parámetro el nivel de dificultad seleccionado y devuelve una matriz numpy que representa el tablero del juego. Las minas deben ser colocadas aleatoriamente en el tablero.
- ***descubrir_casilla(tablero, fila, columna)***: Esta función recibe como parámetros el tablero del juego y las coordenadas de una casilla. Devuelve True si la casilla contiene una mina y el juego debe terminar, o False si no contiene una mina.
- ***colocar_bandera(tablero, fila, columna)***: Esta función recibe como parámetros el tablero del juego y las coordenadas de una casilla. Coloca una bandera en la casilla especificada.
- ***verificar_fin_juego(tablero)***: Esta función recibe como parámetro el tablero del juego. Devuelve True si todas las minas tienen banderas y todas las demás casillas están descubiertas, lo que indica que el juego ha terminado, o False si el juego aún no ha terminado.

Implementa el juego completo utilizando estas funciones y muestra en pantalla el estado del tablero después de cada movimiento del jugador.

Escribe el código completo que cumpla con los requisitos mencionados y proporciona comentarios adecuados para explicar tu lógica y facilitar su comprensión.

Recuerda evaluar y validar correctamente las entradas de los usuarios, así como manejar posibles errores o excepciones que puedan surgir durante el juego.

*Si nunca has jugado al buscaminas revisa el juego en el QR de la primera página. Básicamente los números representan cuántas minas hay vecinas a ese número, en base a eso podrás deducir donde hay minas y colocar las banderas, de esta forma podrás descubrir el resto del tablero y ganar el juego.

UNIVERSIDAD ADOLFO IBÁÑEZ

UNIVERSIDAD ADOLFO IBÁÑEZ

UNIVERSIDAD ADOLFO IBÁÑEZ