Juego 1 - Adivina el número secreto

Descripción:
El objetivo de este ejercicio es desarrollar un juego de adivinanza en Python. El juego consistirá en adivinar un número secreto entre 1 y 10 en un máximo de tres intentos. El programa proporcionará pistas al jugador sobre si el número secreto es mayor o menor que su intento actual.

Instrucciones:

1. Crea un programa en Python que permita al usuario jugar un juego de adivinanza de números.
2. Al comienzo del juego, el programa debe generar un número secreto aleatorio entre 1 y 10 que el jugador debe adivinar.
3. El programa debe solicitar al jugador su nombre.
4. El programa debe dar la bienvenida al jugador e informarle que está pensando en un número entre 1 y 10 y que el jugador debe adivinarlo en tres intentos o menos.
5. Inicia un bucle que permite al jugador adivinar el número secreto. El bucle debe ejecutarse hasta que el jugador adivine correctamente o agote sus tres intentos.
6. En cada intento, el programa debe pedir al jugador que ingrese su suposición.
7. Después de cada suposición, el programa debe proporcionar una de las siguientes respuestas:
    a. Si la suposición es correcta, el programa debe felicitar al jugador y mostrar cuántos intentos se requirieron para adivinar el número. Luego, el juego debe terminar.
    b. Si la suposición es incorrecta y el número secreto es mayor que la suposición del jugador, el programa debe indicar que el número secreto es mayor que la suposición.
    c. Si la suposición es incorrecta y el número secreto es menor que la suposición del jugador, el programa debe indicar que el número secreto es menor que la suposición.
8. Si el jugador no adivina el número secreto en tres intentos, el programa debe informar al jugador que ha agotado sus intentos y revelar el número secreto. El juego debe terminar.

Requisitos Adicionales:

- Asegúrate de que el programa sea claro y fácil de entender.
- Utiliza comentarios para explicar la lógica de tu código.
- Puedes utilizar la función random.randint(min, max) del módulo random para generar el número secreto.
- Puedes utilizar estructuras de control como bucles y condicionales para implementar la lógica del juego.

Juego 2 - El tres en raya

El objetivo de este ejercicio es desarrollar un juego de "Tres en Línea" (Tic-Tac-Toe) en Python. El juego permitirá a dos jugadores competir en un tablero 3x3, tratando de ser los primeros en alinear tres de sus fichas en fila, columna o diagonal.

Instrucciones:

1. Crea un programa en Python que permita a dos jugadores jugar al "Tres en Línea".
2. Diseña un tablero de juego de 3x3 en el que los jugadores puedan realizar sus movimientos.
3. El programa debe solicitar el nombre de los dos jugadores antes de comenzar el juego.
4. El juego debe mostrar el tablero vacío al comienzo, con las casillas numeradas para que los jugadores ingresen sus movimientos de manera más sencilla.
5. Inicia un bucle que permita a los jugadores realizar movimientos alternados. El jugador 1 usará "X" y el jugador 2 usará "O".
6. Después de cada movimiento, el programa debe actualizar el tablero e imprimirlo para que los jugadores vean el estado actual del juego.
7. El juego debe verificar si alguno de los jugadores ha ganado al alinear tres de sus fichas en fila, columna o diagonal. En ese caso, el juego debe declarar al jugador ganador y terminar.
8. Si el tablero se llena y ningún jugador ha ganado, el juego debe declarar un empate y terminar.
9. Si un jugador intenta realizar un movimiento en una casilla ya ocupada, el programa debe mostrar un mensaje de error y permitir que el jugador realice otro movimiento.

Requisitos Adicionales:

Utiliza listas o matrices para representar el tablero del juego. Una posible representación del tablero vacío puede ser esta:

```
[0,1,2]      [X,1,2]      [X,1,2]      [X,1,2]      [X,1,2]
[3,4,5]  →   [3,4,5]  →   [3,O,5]  →   [3,O,5]  →   [O,O,5]  →  …...
[6,7,8]      [6,7,8]      [6,7,8]      [X,7,8]      [X,7,8]
```

Asegúrate de que el programa sea claro y fácil de entender.
Utiliza comentarios para explicar la lógica de tu código.

Ejercicio 3 - Las palabras encadenadas

Descripción:
El objetivo de este juego es desarrollar un programa en Python que permita a los jugadores crear una cadena de palabras en la que cada palabra debe comenzar con las dos últimas letras de la palabra anterior. El juego continuará hasta que uno de los jugadores no pueda encontrar una palabra válida o hasta que decidan finalizar el juego.

Instrucciones:

1. Crea un programa en Python que permita a dos o más jugadores participar en el juego de "Palabras Encadenadas".
2. Al comienzo del juego, solicita a cada jugador que ingrese su nombre. Registra el orden en el que los jugadores participarán.
3. El programa debe comenzar con una palabra inicial, que puede ser proporcionada por un jugador o elegida aleatoriamente de una lista predefinida.
4. Establece un bucle que permita a los jugadores tomar turnos para ingresar una palabra que cumpla con la regla: la palabra debe comenzar con las dos últimas letras de la palabra anterior.
5. El programa debe verificar si la palabra es válida (cumple con la regla) y si ha sido utilizada anteriormente durante el juego. Si la palabra es válida, agrégala a la cadena de palabras y pasa al siguiente jugador.
6. Si un jugador no puede encontrar una palabra válida o si repite una palabra que ya se ha dicho, el juego debe terminar.
7. Muestra un mensaje que declare al jugador que cometió el error como perdedor y finaliza el juego.
8. Puedes proporcionar la opción de reiniciar el juego si los jugadores desean comenzar una nueva ronda.

Requisitos Adicionales:

- Utiliza una lista o estructura de datos para almacenar la cadena de palabras.
- Implementa una función o método que verifique si una palabra es válida y si ha sido utilizada anteriormente.
- Asegúrate de que el programa sea claro y fácil de entender.
- Utiliza comentarios para explicar la lógica de tu código.

Ejercicio 4 - Hundir barcos

Descripción:

El objetivo de este ejercicio es crear un juego en Python llamado "Hundir Barcos". En este juego, los jugadores deben adivinar la posición de los barcos en el agua. Hay tres tipos de barcos: Transatlántico, Acorazado y Submarino, con valores de 3, 4 y 5 puntos respectivamente. Los jugadores tienen tres rondas para adivinar las ubicaciones de los barcos y ganar puntos. El jugador con la puntuación más alta después de 3 rondas gana el juego.

Instrucciones:

1. Crea un programa en Python que permita a dos jugadores competir en el juego de "Hundir Barcos".
2. Al comienzo del juego, el programa debe generar aleatoriamente la ubicación de tres barcos (Transatlántico, Acorazado y Submarino) en una lista de 10 posiciones. Los barcos no pueden superponerse.
3. Los jugadores deben ingresar sus nombres al comienzo del juego para identificarlos.
4. Inicia un bucle que permita a los jugadores tomar turnos para adivinar la posición de los barcos. Cada jugador tiene tres rondas.
5. En cada ronda, el jugador debe ingresar una posición (un número entre 1 y 10) donde cree que se encuentra un barco.
6. El programa debe verificar si la posición ingresada por el jugador corresponde a la ubicación de alguno de los barcos. Si acierta, el jugador gana puntos según el valor del barco (3, 4 o 5 puntos) y el barco se elimina de la lista de ubicaciones.
7. Después de cada intento, muestra un mensaje indicando si el jugador acertó o no, y muestra el puntaje acumulado.
8. El juego debe continuar hasta que se completen tres rondas para cada jugador.
9. Al final de las rondas, muestra el puntaje total de cada jugador y declara al jugador con la puntuación más alta como el ganador del juego.

Requisitos Adicionales:

Utiliza estructuras de datos como listas y diccionarios para representar los barcos, las ubicaciones y el puntaje de los jugadores.
Asegúrate de que el programa sea claro y fácil de entender.
Utiliza comentarios para explicar la lógica de tu código.