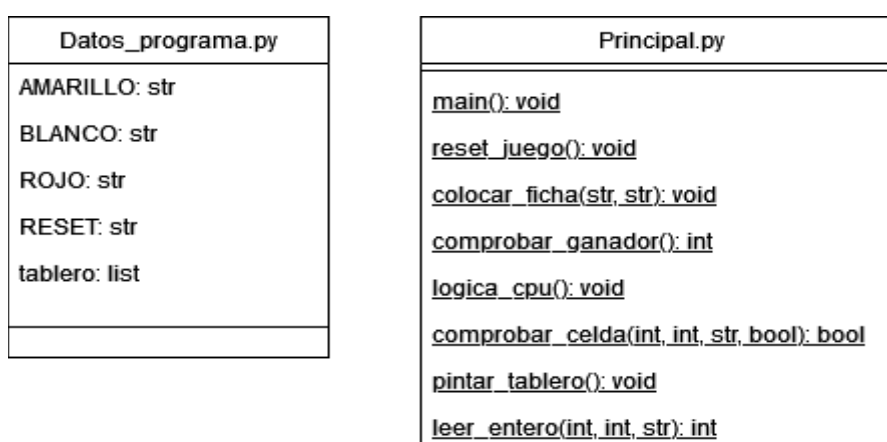


Parte 1.

Crea un módulo de Python que contenga todos los ejercicios creados en las sesiones anteriores. Para ello, crea una función para cada uno de ellos. Debes de controlar las posibles excepciones que se puedan dar.

Parte 2.

Realiza un programa que permita jugar al 3 en raya. Tendrá dos módulos, uno que contiene los datos de configuración del juego y otro con la lógica del mismo. Se pueden ver en el siguiente diagrama:



1. Datos_programa.py: módulo que contiene las constantes del juego y la lista dónde se almacenarán los datos:
 1. tablero: lista bidimensional de cadenas. En él se almacenarán las fichas que ponen los jugadores.
 2. AMARILLO : constante de tipo cadena de caracteres que almacena el código ANSI para pintar un texto de amarillo. Se usará para las X.
 3. BLANCO : constante de tipo cadena de caracteres que almacena el código ANSI para pintar un texto de blanco. Se usará para pintar el resto de tablero y la información al usuario.
 4. ROJO : constante de tipo cadena de caracteres que almacena el código ANSI para pintar un texto de rojo. Se usará para las O.
 5. RESET : constante de tipo cadena de caracteres que almacena el código ANSI para eliminar un modificador de color en el texto.
2. Principal.py: módulo que almacena la lógica del juego. A continuación, se detallan las distintas funciones que contienen:
 1. main: método principal. Debe de contener un menú con las distintas opciones:

0. Salir del juego.

1. Jugador vs CPU: permite que un usuario juegue contra el ordenador.
2. Jugador 1 vs Jugador 2: permite que dos usuarios jueguen.
2. `reset_juego`: limpia el contenido del tablero, dejando una lista que contiene 3 listas con 3 posiciones. Estas almacenarán un espacio en blanco.
3. `pintar_tablero`: función que se encarga de pintar el tablero en pantalla con las fichas que se han colocado.
4. `comprobar_ganador`: recorre los distintos elementos de la lista para ver el resultado de la partida. Devuelve 0 en caso de empate, 1 si ha ganado el jugador 1 y 2 si ha ganado la CPU o el jugador 2.
5. `comprobar_celda`: se encarga de comprobar si una celda tiene ya una ficha o no. En caso que este vacía inserta la ficha que se pasa por parámetro y devolverá True. En caso contrario devolverá False. Recibe los siguientes parámetros:
 1. `fila (int)`: indica la fila de la celda a comprobar.
 2. `columna (int)`: indica la columna de la celda a comprobar.
 3. `simbolo (str)`: ficha que se ha de insertar en la posición indicada en caso de que esté vacía. Podrá ser una X o una O.
 4. `mostrar_mensajes (bool)`: indica si se quiere mostrar un mensaje de error en caso de que la celda esté ocupada.
6. `colocar_ficha`: se encarga colocar la ficha mediante la llamada a la función `comprobar_celda`. Para realizar esta labor utiliza la función `leer_entero` que se explica en el punto 8. Debe de ejecutar la llamada a la función `comprobar_celda` mientras este devuelva un Falso indicando de que no se ha podido colocar la ficha.

Los parámetros que recibe son:

1. `nombre_jugador (str)`: indica el nombre del jugador que se ha de mostrar por pantalla.
2. `ficha (str)`: símbolo que define la ficha de un jugador y que se pasará al método `comprobar_celda`. Puede ser X o O.
7. `logica_cpu`: coloca una ficha al azar cuando le toca jugar al ordenador (debe de llamar a una función que devuelva un método aleatorio) y debe de llamar a la función `colocar_ficha` (descrita en el punto 6) para llevar a cabo la acción.

8. `leer_entero`: función que permite la lectura de un número entero desde teclado que se encuentre entre dos valores. Recibe los siguientes parámetros:

1. `minimo` (int): número mínimo que dará por válido al realizar la lectura.
2. `maximo` (int): número máximo que dará por válido al realizar la lectura.
3. `mensaje` (str): mensaje que muestra por pantalla solicitando que se introduzca un número.

Puedes representar la información por pantalla y el tablero de juego por pantalla como quieras. A continuación, puedes ver una posible representación:

```
-----  
| x |  |  |  
-----  
| o | o |  |  
-----  
|  |  | x |  
-----  
  
Jugador 1:  
Introduzca la fila:
```

En el módulo *principal.py* debes de añadir el siguiente código al final para que se ejecute la función *main*:

```
if __name__ == "__main__":  
    main()
```

Con esta tarea, se está trabajando el RA1, concretamente los CE a y b.