

**PRUEBA TÉCNICA RYC**

**Nombre: Thomas Tellerias**

**Fecha: 29/04/2024**

**Lenguaje utilizado: JavaScript  
Duración: 90 minutos.**

**Responda las siguientes preguntas.**

**“Se puede utilizar pseudocódigo o cualquier lenguaje de programación que sea de su preferencia.**

**Se recomienda utilizar un editor como VS Code y pegar el resultado acá.”**

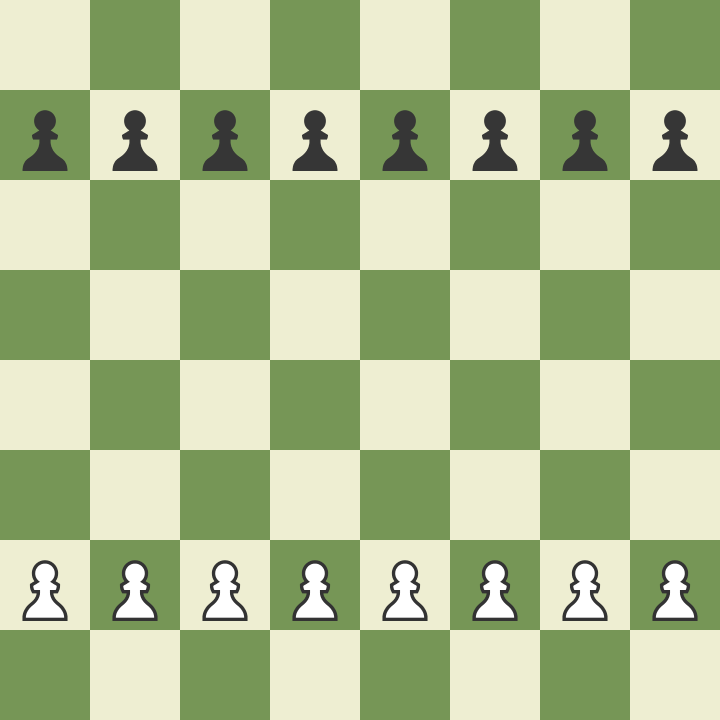
**1.-** Un jugador de ajedrez requiere una función que le permita saber si dado un tablero de ajedrez de entrada (8x8) puede determinar si al mover de posición un peón este:

**a)** Peón Captura a otro Peón (debe retornar el tablero con la nueva posición)

**b)** Movimiento no válido (debe retornar el tablero de entrada sin alterar e indicar que el movimiento no es válido)

**c)** Movimiento válido (debe retornar el tablero con la nueva posición)

**Nota: *El tablero sólo cuenta con 8 peones por jugador.***



**Se requiere:**

**a)** Explicar los parámetros de entrada

**b)** Explicar los parámetros de salida

**c)** Una breve descripción de la lógica que utilizará para resolver el problema

**d)** Satisfacer el requerimiento del jugador

**Importante:** *El peón puede avanzar una o dos casillas en dirección vertical en su primer movimiento, después de adelantado por primera vez solo puede avanzar una casilla, a diferencia del resto de piezas no puede ir hacia atrás y no puede capturar las piezas contrarias que se encuentran en la misma dirección en la que se mueve, solo podrá hacerlo si se encuentran a una casilla en diagonal respecto a él (excepto en la toma de peón al paso).*

**2.-** Cree una función que, dado un número entero permita dibujar un cuadrado utilizando la letra O.

**Se requiere:**

**a)** Explicar los parámetros de entrada

**b)** Explicar los parámetros de salida

**c)** Una breve descripción de la lógica que utilizará para resolver el problema

**d)** Satisfacer el requerimiento

**Ejemplo:** Lado 4



**Respuesta**

1. Los parámetros de entrada: Se utiliza una biblioteca estándar de Node llamada “readline” para obtener un numero entero ingresado por el usuario en consola.
2. Los parámetros de salida: Luego de obtener el valor ingresado por el usuario este es brindado como parámetro a una función que retornara una lista con las líneas a mostrar en consola y dar forma al cuadrado
3. Explicación de la lógica: Basado en el numero entregado por el usuario, se realizan dos ciclos “for” para crear las líneas del cuadrado para ser mostradas en pantalla. En el primer ciclo se inicializa una variable local auxiliar donde se almacenara una línea del cuadrado para luego ser agregada a una lista y ser nuevamente inicializada hasta acabar el ciclo. Dentro del segundo ciclo se comprueba que no se trate de la primera línea ni de la ultima para saber si almacenar espacios, que irían dentro del cuadrado.
4. Imagen del código:

Texto

Descripción generada automáticamente

Prueba en consola:

Texto

Descripción generada automáticamente

**3.-** Cree una función que permita validar que una expresión con paréntesis, llaves y corchetes se encuentra equilibrada (significa que paréntesis, llaves y corchetes de una expresión se abren y cierran en orden de forma correcta).

**Ejemplo:**   
 Expresión equilibrada: { [ a \* ( c + d ) ] – b }

Expresión no equilibrada: { a \* ( c + d ) ] – b }

**Se requiere:**

**a)** Explicar los parámetros de entrada

**b)** Explicar los parámetros de salida

**c)** Una breve descripción de la lógica que utilizará para resolver el problema

**d)** Satisfacer el requerimiento

**Respuesta**

1. Los parámetros de entrada: se le pide al usuario que ingrese en la consola una expresión con paréntesis, llaves y/o corchetes
2. Los parámetros de salida: Se muestra la expresión ingresada por el usuario nuevamente luego de ser analizada y se muestra si es valida o no valida.
3. Lógica para resolver el problema: Luego de que el usuario ingrese una expresión, una función la recibe como parámetro para recorrer carácter por carácter dicha expresión, contando los paréntesis, llaves y corchetes tanto de apretura como de cierre para luego comparar que estos estén equilibrados y retornar ‘true’ si se encuentra equilibrada y ‘false’ en caso contrario
4. Imagen del código:

Texto

Descripción generada automáticamente

Prueba en consola:

Texto

Descripción generada automáticamente

**4.-** Piedra (R), Papel (P) o Tijera (S)

Se requiere crear una función que permita determinar quien es el ganador de un mejor de 5 (Un jugador requiere de 3 victorias para ganar) para el juego de “Piedra, Papel o Tijera”.

Para resolver este problema se requiere tener en cuenta lo siguiente:

1. Como entrada se recibirá un objeto JSON con la siguiente estructura

{

jugador1: [“R”, ”P”, ”P”, ”S” , ”R”],

jugador2: [“P”, ”P”, ”P”, ”R”, ”P”]

}

1. Para que se pueda comprobar si existe un ganador ambos jugadores deben haber realizado sus 5 jugadas.
2. Empate es una respuesta valida.
3. Como respuesta se espera: Victoria Jugador X o Empate.

**Se requiere:**

**a)** Explicar los parámetros de entrada

**b)** Explicar los parámetros de salida

**c)** Una breve descripción de la lógica que utilizará para resolver el problema

**d)** Satisfacer el requerimiento

**Respuesta**

1. Los parámetros de entrada: se recibe un json con dos listas de strings, donde cada string corresponde a una tirada en el juego de cada jugador
2. Los parámetros de salida: Dependiendo de quien consiguió mas victorias dentro de las 5 tiradas correspondientes se muestra en consola si gano el jugador 1 o el 2, o si es que concluye en un empate el juego
3. Lógica del programa: Se recorren las listas con índice de referencia en común, comparando los resultados de cada tirada y contando las victorias y sumándolas a cada jugador y dependiendo de quien consiguió mas se muestra por consola al ganador
4. Imagen del código:

Texto

Descripción generada automáticamente con confianza media

Prueba en consola:

Entrada:

Texto, Word

Descripción generada automáticamente con confianza media

Output:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente