



Cátedra Tecnología de Sistemas

[Introducción a la Programación]

Código: [00831]

Tarea 2. Valor 2%

Temas de Estudio

1. Tema 1 y Tema 2 vistos anteriormente
2. Tema 3. Subtemas
 - a. Funciones y una introducción a la recursividad
 - b. Plantillas de clase array y vector; captura de excepciones Instrucciones de control;

Objetivo

Resolver un problema, con un programa en el lenguaje de C++ aplicando lo aprendido en los temas de estudio

Software de Desarrollo

CodeBlocks, en la plataforma MOODLE está disponible las instrucciones para su instalación

Desarrollo

Se debe de desarrollar un programa que simule el juego de **Conecta 4**.

Conecta 4 (Cuatro en Fila) es un juego divertido para niños, jóvenes y adultos. El objetivo del juego es conectar 4 fichas de tu mismo símbolo en fila, ya sea horizontal, vertical o diagonal antes que tu oponente lo haga. Juega contra la computadora o juega contra un amigo en el mismo dispositivo.

Juego Conecta 4

1. Inscribir jugador
2. Jugar partida
3. Jugar partida contra PC
4. Resultados

5. Salir del Programa

Importante: El menú debe repetirse siempre hasta que se ingrese el valor de salida #5

Inscribir jugador

Debe solicitar # de cédula, solo el número sin guiones y de 9 dígitos. Adicional el nombre del jugador.

Jugar partida y jugar partida contra PC

En esta opción debe presentar el tablero con el que se va a jugar, que se visualizaría de la siguiente forma:

1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-
	1	2	3	4	5	6	7	8

La diferencia entre la opción *Jugar Partida* y *Jugar Partida contra PC* es, en la primera debe preguntar a cada jugador (total de jugadores 2) el valor para cada turno. En el caso de la opción *Jugar Partida contra PC* solo pide el del jugador y la máquina aleatoriamente jugará. No es requerido crear una estrategia de juego para la PC, solamente debe generar aleatoriamente un valor de juego, en un espacio donde tenga otra ficha hacia abajo y no esté lleno de C o T, cuando sea el turno de la PC. Es probable que siempre gane el jugador a la máquina.

En cada turno debe solicitar fila y columna donde quiere poner el valor, los valores o símbolos que se van a utilizar son **C** y **T**

Ejemplo de mensajes requeridos:

Ingrese el # de cédula del jugador: #####

Jugador valido, Bienvenido al juego
Jugador inválido, intente nuevamente

Turno del jugador #1

Indique la fila y columna donde jugará: #####

Turno del jugador #2

Indique la fila y columna donde jugará: #####

Felicidades jugador # ha ganado la partida

Presione S si desea regresar al menú principal (S/N):____

Juego termina sin ganador cuando la tabla este llena de C y T, sin alternativas disponibles.

Resultados

Debe mostrar en esta opción una tabla de resultados por jugador

Ejemplo opción #4:

Resultados Conecta 4

Cédula	Nombre	Cantidad de partidas ganadas
#####	Jugador #1	10

Desea regresar al menú principal (S/N):____

Codificación:

1. En el menú deben aparecer solo las opciones indicadas, no debe omitir ninguna ni agregar opciones. En caso de que el usuario digite un valor no existente debe mostrar el mensaje de : **“Opción inválida, vuelva a intentarlo”**
2. La información del Jugador debe almacenarse en una matriz de 10 X 3 posiciones, eso quiere decir, que solamente se podrán inscribir 10 jugadores diferentes. En cada fila de la matriz se almacena: Cédula de Identidad (int), Nombre Jugador (string) y Cantidad de ganec (int).

3. Antes de iniciar una partida debe solicitar jugador o jugadores a participar por número de cédula. Si el jugador no esta inscrito debe presentar mensaje de que no esta inscrito. Una vez validado el jugador, debe asignarle un símbolo, ya sea **C** ó **T** a cada jugador.
4. El tablero debe ser una matriz de 8 X 8, que inicialmente se debe inicializar con el valor "-".
5. Debe generar una función que evalúe si en el tablero se tiene 4 valores de tu mismo símbolo (C o T) en fila, ya sea horizontal, vertical o diagonal. Esta evaluación la debe hacer en cada turno. Si identifica que hay 4 valores, declara ganador y termina la partida.
6. En la opción de resultados, lo que debe presentar es la matriz de jugadores.
7. Para cada jugador debe acumular la cantidad de ganec de todas las partidas sin que sean necesariamente consecutivas. Los ganec no necesariamente son consecutivos, es decir, jugador #1 puede jugar varias partidas intercaladas con otros jugadores y de igual forma acumula cada gane.
8. Cada valor ingresado se debe validar, por ejemplo:
 - a. si el jugador ingresa más de la cantidad de filas y columnas existentes,
 - b. si ingresa una fila y columna que ya cuenta con un símbolo (C o T)
 - c. si ingresa un valor diferente a S/N cuando corresponde
 - d. si la cédula no es un número
 - e. los valores del menú, etc.Debe solicitar el valor hasta que este correcto.
9. En los turnos de partida siempre debe presentar el tablero, y el mismo debe ir presentando los valores de C y T que se vayan solicitando.
10. Las validaciones de fuera de rango de la matriz debe realizarlo vía excepción, utilizando TRY-CATCH-THROW , presentando un mensaje de error coherente con la validación, si es que se ingresa más de 10 jugadores, debe indicar que se llegó al límite de jugadores. Si es que el usuario ingresa una fila y columna fuera del tablero también.
11. El menú debe repetirse siempre hasta que se ingrese el valor de salida #5. Es decir, la única forma de que el programa termine debe ser que el jugador vuelve al menú desde donde este y presiona la opción #5 para Salir.

Honestidad Académica



<https://audiovisuales.uned.ac.cr/play/player/23048>

Nota Importante

Cada estudiante es responsable del contenido que entrega, si no es el archivo correcto, no podrá entregarlo posterior a la fecha establecida.

Si el contenido del archivo coincide con algún otro estudiante, o se comprueba que no es de su autoría, se aplicaría lo indicado en la plataforma en el documento [Lineamientos ante casos de plagio](#)

Indicaciones Importantes

- Es obligatorio que incluya todo el directorio donde se encuentra < nombre del instrumento>.
- La < nombre del instrumento > debe estar desarrollado en [IDE de desarrollo] que es la herramienta oficial del curso.
- El programa debe ser modular, utilizando de la mejor manera funciones definidas por usted.
- Los trabajos deben realizarse en forma individual. Dentro del código del programa debe de indicar la documentación que explique cómo fue realizado el programa.
- Si utiliza código de algún ejemplo del libro, o de otra fuente que no sea de su autoría, debe de indicarlo.
- Comprima todos los archivos en un solo archivo .zip o .rar.
- **Nombre del archivo que envía:** debe ser nombre y primer apellido del estudiante, y nombre de la tarea. **Ejemplo: JuanRojas-tarea1.**
- La entrega de la <Nombre del instrumento> en las fechas establecidas en la plataforma de aprendizaje en línea Moodle en el apartado que se indique.



Si no concluyó a tiempo la tarea, debe entregar lo que pudo hacer e incluir una carta explicando las razones por las cuales no finalizó.

Rúbrica de Evaluación

Criterio	Cumple a satisfacción lo indicado en la evaluación	Cumple medianamente en lo indicado en la evaluación	Cumple en contenido y formato pero los aportes no son significantes	No cumple o no presenta lo solicitado
Formato: Nitidez y presentación del código, incluyendo Redacción - Ortografía // Documentación interna dentro del código	5	3	1	0
Orden y claridad en el planteamiento (lógica). Cómo ordena las ideas para lograr la mejor solución, aplicando correctamente los conocimientos y herramientas vistos hasta el momento en el curso. Logra completar la funcionalidad solicitada en el enunciado.	5	3	1	0
Estructuras de control – Secuenciales. Utiliza <i>if</i> , <i>if/else</i> y <i>switch</i> en la solución de forma adecuada. Es evalúa que su uso en la lógica sea correcto. Ejemplo: Menús, Preguntas, validaciones de datos, evaluación de datos, identifica correctamente ganador en la matriz, jugador inscrito, etc. .	15	10	5	0.1
Estructuras de control - iterativas. Utiliza <i>while</i> , <i>do/while</i> y <i>for</i> en la solución de forma adecuada. Se evalúa si alguna estructura se encicla por la lógica errónea utilizada. Ejemplo: recorrido de números, validaciones, matrices, etc.	15	10	5	0.1
Manejo de Excepciones. Utiliza las necesarias y	5	3	1	0.1

suficientes adicionales a la o las solicitadas en el enunciado. Con un correcto nombre, parámetro (si aplica) y llamado de las mismas				
Funciones Solicitadas. Creación de la función para el <i>llenado de la matriz, mostrar la matriz y validación de matriz si hay ganador</i> según se solicita en el enunciado del problema.	15	10	5	0.1
Funciones Adicionales. Creación de al menos 2 funciones más para estructurar la lógica de la solución.	15	10	5	0.1
Arreglos. Correcta declaración, recorrido y uso en general de los arreglos solicitados o propios, en la solución planteada. Tomando en consideración cualquier solicitud específica en el enunciado.	10	5	2	0
Impresión de información en pantalla (Calidad-validez datos/presentación tabulada). Uso correcto de entrada y salida de datos por pantalla. Solicitud de información, validaciones, presentación de lo mínimo solicitado	10	5	2	0
Interfaz de usuario en general. (NO GUI) - aplicación fácil usar e intuitiva. Se refiere a evaluar la distribución y uso de la pantalla, menús y dinámica de uso de la solución. Incluyendo lo mínimo según el ejemplo del enunciado o lo adicional que el estudiante entienda necesario.	5	3	1	0
TOTAL	100			