

**Instrucciones:**

Proceda a resolver el ejercicio propuesto, puede utilizar el SDK Visual Studio Code con el compilador de C++ gcc/g++, o bien el de su elección, y proceda a subir únicamente los archivos de código fuente (\*.cpp/\*.h) a la plataforma de mediación virtual.

Fecha límite para la entrega: **07 de julio 11:50 pm.**

Este proyecto puede realizarlo individual o en parejas si lo desean, cada uno debe subir una copia a mediación antes de la fecha y hora, así como indicar el nombre y carné de ambos.

**Detalle del problema:** Diseñe y programe en C++, usando el entorno de desarrollo de su elección, un aplicativo estilo **juego versus** con las siguientes características:

El juego consiste en un enfrentamiento entre dos procesos/jugadores que compiten en un mismo tablero de juego, y cada jugador es un proceso independiente (hilo) ejecutado en paralelo.

Internamente el sistema debe implementar una matriz de 10 x 10 espacios, que pueda almacenar números enteros. La matriz debe llenarse de manera aleatoria con números del 0 al 9 inclusive, los números impares le corresponden al jugador/proceso número 1 y los números pares al jugador/proceso número 2 (incluido el 0). La matriz debe contener igual número de valores pares que impares (debe contar los valores pares e impares y en caso de no ser iguales puede cambiar valores por par o impar según sea el caso, o bien, volver a generar los números hasta que obtenga igual número). La matriz es el tablero de juego, y ambos procesos acceden y modifican la misma matriz.

**Mecánica del juego:**

El juego consiste en que cada jugador (proceso) va a ir “comiendo” espacios de la matriz, con la validación que el espacio seleccionado le corresponde correctamente a cada jugador sin “caer” o seleccionar un espacio del contrario.

Cada proceso puede “moverse y avanzar” en la matriz de manera aleatoria, obteniendo dos valores, uno para la fila y otro para la columna. Cada jugador se mueve en su propio proceso, pero deben esperar 2 segundos en cada selección de espacio para “probar suerte”, o sea cada dos segundos es que van jugando.

Mecánica del juego:

- **Proceso jugador 1** (que le corresponden los números impares): selecciona aleatoriamente el espacio [2],[6] por ejemplo (debe imprimir en pantalla ambos valores “coordenadas” indicando que es el espacio seleccionado por el jugador 1, así como el valor de la matriz correspondiente), y en caso de ser un número impar se marca la casilla del tablero (espacio de la matriz) como “usado”, por ejemplo, un -1, y espera dos segundos antes de seguir.
- **Proceso jugador 2** (que le corresponden los números pares): selecciona aleatoriamente el espacio [5],[9] (debe imprimir en pantalla ambos valores “coordenadas” indicando que es el espacio seleccionado por el jugador 2, así como el valor de la matriz correspondiente), en caso de ser un número par se marca la casilla del tablero (espacio de la matriz) como “usado”, por ejemplo un -2, y espera dos segundos antes de continuar.
- El valor “**usado**” puede ser un comodín, como ejemplo un valor negativo como -1 si es un valor seleccionado previamente por el jugador/proceso 1, -2 si es un valor seleccionado previamente por el jugador/proceso 2 ó -3 si es un espacio seleccionado erróneo.

- Cuando un jugador/proceso selecciona una posición de la matriz con un número que no le corresponde, la misma se marca como “usado” (por ejemplo -3) y el jugador/proceso que incurre en la falta se le suma una penalidad, esto indica que debe llevar un conteo de penalidades por jugador.
- El juego termina cuando el jugador/proceso X **consume todos los espacios del tablero** de juego (la matriz) que le corresponde (jugador 1 valores impares y al jugador 2 valores pares), o bien, un jugador comete 10 faltas, en cuyo caso el ganador es el otro jugador.
- El proceso debe indicar si termina ganando o bien si termina ya que cometió 10 errores.
- Si un jugador/ proceso selecciona un espacio de la matriz que ya fue seleccionado previamente, se debe validar lo siguiente:
  - Si el espacio seleccionado fue marcado como exitoso (-1 en caso de jugador uno, o bien, -2 en caso de jugador dos) y se selecciona por el proceso contrario (ejemplo -1 por el dos / -2 por el uno) se le atribuye una penalidad.
  - Si el espacio seleccionado fue marcado como erróneo (-3) el proceso/ jugador que lo seleccionó lo debe reclamar cambiando el valor de -3 por -1 en caso de ser seleccionado por el jugador1, o -2 en caso de ser seleccionado por el jugador 2.
  - Si el espacio seleccionado fue marcado exitosamente por él mismo antes (-1 por el proceso uno y /o -2 por el dos), no pasa nada, simplemente se continua la ejecución del juego.
- Cuando ambos procesos terminan, el proceso principal debe imprimir la matriz original (la generada antes de empezar el juego), así como la matriz luego del juego (con los indicadores/comodín de selección de espacios, o sea -1, -2, y/o -3) y cuál jugador/proceso ganó o bien, perdió de último (tuvo 10 errores primero).

### Criterios de evaluación:

Para el desarrollo de esta tarea, puede usar cualquiera de los conceptos estudiados durante el semestre, como objetos, librerías, manejo de errores, lectura y escritura de archivos. Debe seguir las métricas descritas a continuación:

1. 15% Uso de clases y objetos.
2. 15% Manejo de errores.
3. 25% Implementación de estructuras de datos adecuadas.
4. 25% Lógica y desarrollo del problema.
5. 20% Orden, documentación interna, buenas prácticas.