# Práctica 4 - Adicional

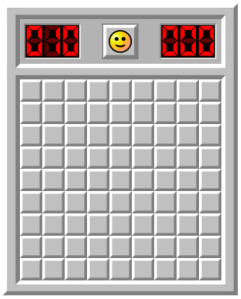
## Tipos de datos compuestos – Estructuras de datos dinámicas

Aclaración: Esta práctica tiene por objetivo repasar y practicar el tipo de datos struct, funciones, parámetros, matrices y recursión. La idea es repasar los conocimientos adquiridos por aquellos alumnos que están adelantados siempre y cuando no se retrasen con las prácticas que vendrán. Por este motivo esta es una práctica adicional.

1. Incorpore al ejercicio 6 de la práctica 4 las siguientes funciones. Modifique la estructura y reimplemente las funciones según corresponda:
   a. Insertar un elemento de forma ordenada.
   b. Reimplementar la función que calcula la cantidad de elementos de forma eficiente (sin recorrer la lista). Analice y modifique todas las funciones necesarias.

2. Utilizando la estructura y funciones del ejercicio anterior escriba un programa que lea números enteros positivos desde teclado hasta ingresar el número 0. Los números leídos deben ser almacenados en orden ascendente en una lista enlazada. Generar 2 listas nuevas con los números pares e impares. Imprima las 3 listas junto con la cantidad de elementos. Por último, libere la memoria reservada dinámicamente.

3. **Juego del Buscaminas**:



Desarrollar una aplicación en modo texto del viejo juego Buscaminas (en ingles: Minesweeper). Este es un videojuego para un jugador inventado en 1989 que se hizo popular desde que Microsoft Windows lo incluyo en su versión de Windows 3.1.

El objetivo del juego es identificar todas las bombas que se encuentran ocultas en cada una de las casillas del tablero sin "pisarlas" o descubrirlas. El jugador debe ir explorando el tablero descubriendo las casillas hasta que solo queden ocultas las bombas para ganar el juego. Si en el camino selecciona una casilla con una bomba el juego finaliza y el jugador pierde.

Al descubrir una casilla que no tiene bomba, el juego muestra el número de minas que hay en las casillas circundantes o vecinas. Por ejemplo, si una casilla tiene el número 3, significa que de las 8 casillas que hay alrededor (si no es en una esquina o borde del tablero) hay 3 con minas y 5 sin minas. Si se descubre una casilla sin número indica que ninguna de las casillas vecinas tiene mina y estas se descubren automáticamente para ahorrarle el trabajo al jugador.

Para ayudar al jugador, el juego permite marcar (y desmarcar) una casilla sospechosa de contener una bomba.

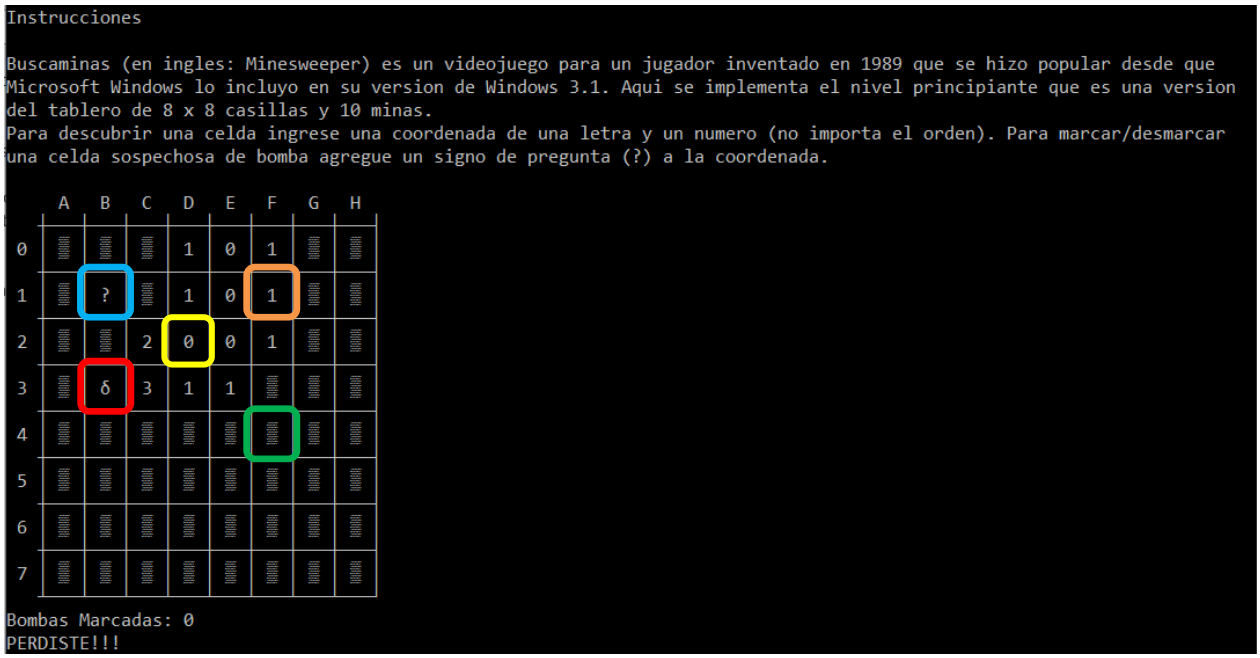
Hay muchas implementaciones disponibles en línea, una de ellas se puede acceder en el sitio https://minesweeper.online/es/. Se recomienda que dediquen unos minutos para jugar algunas partidas con el objetivo de comprender la dinámica del juego.

Para este ejercicio debe implementar el nivel principiante que es una versión para un tablero de 8 x 8 casillas y 10 minas. Básicamente, el juego consiste en la siguiente secuencia de pasos que se repiten hasta que se termine:

- *Mostrar el estado actual del tablero:* se refleja el estado individual de cada casilla que puede estar oculta (verde), oculta y marcada como posible bomba (celeste), visible sin bomba con el número de bombas alrededor (naranja) o visible con bomba (roja, juego perdido).
- *Realizar una jugada:* una jugada consiste en una coordenada de una letra de la A a la H y un número del 0 al 7. Además, puede indicarse de forma opcional el signo de pregunta (?) para marcar o desmarcar una casilla sospechosa de bomba sin mostrarla.
- Actualizar el estado de juego: en función de la jugada realizada se visibiliza o marca una casilla.

Para visualizar una casilla hay que tener en cuenta que:

- Si en la casilla hay una bomba se muestra un carácter especial y el juego se termina (recuadro rojo en imagen).
- Si no hay una bomba (al seleccionar la casilla) se muestra el número de bombas alrededor de la casilla (entre 0 y 8, el 0 podría reemplazarse por un carácter en blanco).
- En el caso de que se visualice una casilla que no tiene bombas alrededor (recuadro amarillo en la imagen) deben mostrarse automáticamente las casillas vecinas hasta encontrar una casilla que tenga bombas alrededor
- Para marcar una casilla se muestra un signo de pregunta (?), mientras que para desmarcarla se vuelve a mostrar un carácter especial para indicarlo (recuadro celeste en imagen).




El juego finaliza cuando se realiza una jugada que muestra una casilla que contiene una bomba (el jugador pierde) o cuando el jugador visibiliza todas las casillas menos las bombas (el jugador gana).
Tenga en cuenta los siguientes consejos para desarrollar con mayor facilidad el juego:

- Utilice constantes siempre que tenga oportunidad: tamaño del tablero, cantidad de casillas, caracteres visualizables, estado del juego, etc.
- Utilice una estructura para representar una casilla. Sería conveniente almacenar el estado (visibilidad, bomba, marca, bombas en la vecindad).
- Desarrolle la aplicación de manera modular, utilizando un buen número de funciones.
- No pierda tiempo en mostrar el tablero de una manera agradable. Modularice con una función lo más simple posible y una vez resuelto el problema vuelva a implementarlo. Recuerde que puede realizar un *for* sobre una variable *char* imprimiendo los caracteres ASCII especiales (en el demo se usaron los códigos hexadecimales 0xB1 para casilla oculta y 0xEB para la bomba).
- Es conveniente tener calculado la cantidad de bombas vecinas para cada casilla una vez inicializado el tablero.
- Mientras desarrolle no cambie la semilla de generación de números aleatorios. De esta manera siempre se obtiene la misma secuencia, algo sumamente útil para realizar pruebas.
- Para mostrar la vecindad de aquellas casillas que no tienen bombas es conveniente utilizar un algoritmo recursivo. Deje esta funcionalidad para desarrollar luego de implementado el juego. Analice bien la condición de corte y la forma de llamada recursiva.