

Ejercicio de Práctica en Papel

Nombre y Apellido:

El siguiente ejercicio debe ser resuelto en esta hoja con lapicera/lápiz. Puede utilizar hojas auxiliares. Todas las hojas deben tener nombre y apellido. Tener en cuenta la legibilidad de la solución entregada, ya que de no comprenderse lo escrito no se podrá corregir. Deben entregarse todas las hojas que contengan código asociado a la solución. Recuerde que si hace uso de funciones auxiliares, debe incluirlas y/o codificarlas según corresponda.

Tiempo de Resolución: 90 minutos.

Puntaje Requerido: 24/40 puntos.

Consigna: en el barrio cerrado **19022025**, cada familia de las 100 que viven ahí, poseen un lote cuadrado de 30 mts por 30 mts. Los lotes están distribuidos en 10 filas x 10 columnas. Como el barrio está en construcción y será muy tranquilo (no habrá problemas de seguridad de ningún tipo), los vecinos tienen que ponerse de acuerdo en cuestiones muy importantes:

- **Familia:** un string
- **Cantidad de Perros:** cantidad de perros en el lote (en 0..3).
- **Árboles:** cantidad de árboles en el lote (2..10).

Las cantidades anteriores fueron aprobadas por todos los vecinos del barrio cerrado (Ud. puede estar seguro que no se manejan valores fuera de rango).

La información relevante de cada lote se guarda en un **TDA FPA** (de "Familia, Perros y Árboles), el cual Ud. debe diseñar y codificar las funciones necesarias para asignarle y recuperar los valores de los campos de **FPA**.

Toda la información del proyecto se encuentra almacenada en una lista enlazada **LOTES**, que contiene la información de cada lote ya vendido del proyecto (en esta lista no hay información de los lotes todavía no vendidos, para ahorrar espacio). Para cada **NODO** (un lote) se guarda la siguiente información:

- Coordenadas: fila y columna del lote (en 1..10).
- Info: la información del TDA FPA para el lote.

LOTES se encuentra ordenada por filas y columnas.

En base a la descripción anterior, se solicita:

a) Defina todas las estructuras de datos necesarias, con el formato que considere apropiado para representar un tipo de datos **NODO** y la lista **LOTES** que está constituida por nodos de este tipo. Declare la variable **A**, del tipo nodo, y asígnele los siguientes valores: fila 1, columna 1, familia "Gonzalez", perros 2 y árboles 6.

b) Codifique la función *VerdorAmigable()* que recibe la lista **LOTES** y un número de fila, y retorna el grado de "amigabilidad con el medio ambiente" de dicha fila, el cual se calcula en base a:

- cantidad de árboles en la fila / cantidad de perros en la fila

Si el contenido de la lista **LOTES** es:

(1,1,2,5) -> ... -> (2,1,2,4) -> (2,4,3,6) -> (2,8,2,4) -> (3,4,8, 8) ->....

La llamada a *VerdorAmigable(Lotes, 2)* retornará 2.0

c) Implemente una función recursiva *ColumnaSilenciosa()* que reciba la lista **LOTES** y un número de columna, y retorne true si la cantidad total de perros en los lotes ubicados en dicha columna, es igual a 0.

d) Implemente una función *LotesDisponibles()* que reciba la lista LOTES y retorne una lista enlazada **LIBRES** que contenga en cada elemento las coordenadas de cada uno de los lotes libres y un valor entero X, el cual será:

- 1: si el lote está libre y sin vecinos adyacentes en la misma fila.
- 2: si el lote está libre en una columna silenciosa
- 3: si el lote está libre.

La lista LIBRES debe quedar ordenada de menor a mayor según el valor de X.

e) Para las funciones *ColumnaSilenciosa()* y *LotesDisponibles()* expresar y justificar la complejidad de las mismas.

Importante: Para la resolución del problema el alumno puede codificar todas las estructuras de datos y funciones que considere necesarias. Los campos de las estructuras deben respetar lo enunciado en la consigna. En los casos en los que no se indica un prototipo, los parámetros formales de las funciones (cantidad y tipo) deben definirse según los objetivos propuestos. El puntaje final obtenido tendrá en cuenta la eficiencia de la estrategia de resolución elegida.