La asociación de vecinos de la urbanización en la cual usted vive requiere de su colaboración para construir un sistema en Java que permita acceder y administrar los datos de los vecinos. A continuación se describen los requerimientos de dicho sistema.

Requerimientos:

1. Se requiere guardar la información de los edificios que conforman la urbanización, para lo cual se necesita implementar la clase EDIFICIO con al menos el campo NOMBRE y la lista de apartamentos. Usted podrá implementar los métodos de esta clase que crea necesarios.
2. Por cada vecino (clase VECINOS), se requiere saber el nombre, la cédula de identidad, el teléfono, edificio y apartamento. Tome en consideración que en un apartamento pueden vivir varios vecinos. Usted puede agregar los métodos que crea conveniente.
3. Por cada apartamento, se deberá conocer, al menos, el número del mismo y la lista de sus habitantes.
4. Se requiere la implementación de la clase ASOCIACIÓN\_VECINOS, la cual tendrá como atributos, únicamente una lista de edificios y una lista de vecinos. La lista de vecinos debe estar ordenada por cédula de identidad. Además, esta clase deberá contener los siguientes métodos:
   1. **Función buscar\_vecino ((E) entero CI): \* vecino**;  la cual permitirá encontrar la dirección de memoria correspondiente al nodo que contiene los datos del vecino, en el caso de que exista. Esta función devuelve un apuntador a un objeto de la clase vecino y debe tener una complejidad de O(n).
   2. **Función buscar\_Apto\_vecino((E) vecino\* vec): \* apartamento**; la cual permitirá conocer el apartamento de un vecino. Se le pasa como parámetro un apuntador a un objeto de la clase vecino y devolverá un apuntador a un objeto de la clase apartamento. Esta función debe tener una complejidad de O(1).
   3. **Función buscar\_Edificio\_apartamento((E) apartamento\* apto): \* edificio**; que permitirá encontrar el edificio al cual pertenece el apartamento, cuya dirección de memoria es guardada por apto. Esta función debe tener una complejidad de O(1).
   4. **Procedimiento buscar\_dirección\_vecino ((E) entero CI)**; que imprimirá la dirección de un vecino, dada la cédula de identidad. Su complejidad deberá ser de O(1).
   5. **Procedimiento cargar\_Datos()**: que leerá los datos contenidos en un archivo de texto que será entregado por el profesor y los cargará en la estructura de datos que usted haya construido.
   6. No está permitido modificar los parámetros de los procedimientos y funciones antes indicadas. Sin embargo, usted podrá agregar los métodos que crea necesarios para la clase ASOCIACIÓN\_VECINOS.
5. Usted deberá utilizar listas implementadas como apuntadores (implícitos) en Java para resolver el problema planteado. Además, usted podrá agregar los atributos que crea necesarios en las clases de las cuales se requiere generar listas. También puede agregar clases además de las ya indicadas de ser necesario.
6. La estructura de datos a diseñar para este sistema debe tomar como principio la ausencia de redundancia de información; es decir, no se debe repetir información entre los objetos. Por ejemplo, la clase VECINO no debe contener los atributos EDIFICIO y APARTAMENTO, porque esto generaría varios objetos de tipo VECINO con la misma información en dichos campos, ya que muchos vecinos pertenecen a un mismo edificio y varios vecinos pueden estar en mismo apartamento.
7. Deberá utilizar las primitivas de la clase lista vistas en clase. Tome en cuenta que debe mantener la cohesión en las clases a utilizar.