Entregas de paqueterías

*Definición del contexto*

Ud. Y su equipo deben confeccionar un sistema para gestionar la entrega de paquetería de una empresa de correo en el territorio nacional. Disponen de dos archivos binarios uno **Camiones** ordenado por **IdCamion** que lo relaciona con una zona de reparto. Cada zona está constituida por un conjunto de códigos postales, por lo que un CP está asignado a un único vehículo. El archivo esta ordenado por **IdCamion** con repetición de clave (una por cada CP, sin orden por estos)

|  |  |
| --- | --- |
| IdCamion | int[1..100] |
| CP | Int |

El otro archivo **Envios** es el de envíos del dia ordenados por HH:MM de entrega, cada registro con:

|  |  |
| --- | --- |
| codigoPaquete | Char[25] |
| direccion | Char[50] |
| CP | Int |
| IdCamion | int |
| horaEntrega | HH:MM |

Se requiere, para optimizar el procesamiento, llevar los datos a memoria. La sugerencia del líder del equipo es utilizar un vector (con una posición para cada camión), con un puntero del que se desprenda una lista ordenada por CP, la que contenga, a su vez, una subestructura que permita colocar en el orden en que aparecen los paquetes en el archivo correspondiente. La sugerencia en este caso es una cola.

*Se pide*

1. (1 punto) Declarar todas las estructuras de datos necesarias para administrar los archivos y las estructuras en memoria.
2. (1 punto) Desarrolle el prototipo de la función **leerVehiculos** que procese el archivo de **Camiones** y genere el vector de listas (Camion- Codigo postal) sugerido.
3. (2 puntos) Desarrolle la función descripta en el punto anterior
4. (3 puntos) Desarrolle la funcion **planificarEntrega**s que procese el archivo de **Envios** almacene en la estructura de los camiones y según el código postal los paquetes en el orden en el que se encuentran en el archivo.
5. (3 puntos) Desarrolle la función **entregarPaquete**(int, int) que reciba una idCamion, un código postal y quite el próximo paquete pendiente de entrega.

*Dispone de las siguientes funciones de biblioteca puede usar sin desarrollar y para distintos tipos de dato*

void queue(Nodo\* &inicio, Nodo\* &fin, Tipo x), Tipo unQueue(Nodo\* &inicio, Nodo\*&fin) o sus sinónimos

Nodo\* insertarSinRepetir(Nodo\* &Lista, Tipo x), Nodo\* insertarOrdenado (Nodo \*&Lista, Tipo x)

Nodo\* buscarEnLista (Nodo\* lista, Tipo x) 🡪 Esta? Dirección de memoria : NULL