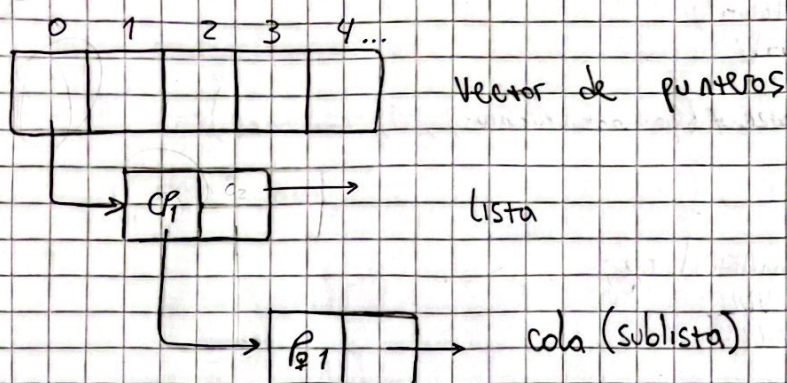


Final sin fecha



1.

```
struct CamionEnArchivo{
```

```
    int idCamion; // [1... 100]
```

```
    int CP;
```

```
};
```

```
struct EnvioEnArchivo{
```

```
    char codigoPaquete[25];
```

```
    char direccion[50];
```

```
    int CP;
```

```
    int idCamion;
```

```
    int horaEntrega;
```

```
};
```

```
struct Paquete{
```

```
    char codigoPaquete[25];
```

```
    char direccion[50];
```

```
    int horaEntrega;
```

```
};
```

```
structCodigoPostal{
```

```
    int CP;
```

```
    NodoSub * frente;
```

```
    NodoSub * fin;
```

```
};
```

```
struct NodoSub{
```

```
    Paquete info;
```

```
    NodoSub * siguiente;
```

```
};
```

```
struct Nodo{
```

```
    CodigoPostal info;
```

```
    Nodo * siguiente;
```

```
};
```

```
Nodo * vCamiones[100];
```


2 y 3.

```
void leerVehiculos(FILE* camiones, Nodo* vCamiones[]) {
```

```
    CamionEnArchivo camionActual;  
    CodigoPostal cpActual;
```

```
    fread(&camionActual, sizeof(CamionEnArchivo), 1, camiones);
```

```
    while (!feof(camiones)) {
```

```
        cpActual.cp = camionActual.cp;  
        cpActual.frente = NULL;  
        cpActual.fin = NULL;
```

```
        insertarSin Repetir (vCamiones[camionActual.idCamion-1], cpActual);
```

```
        fread(&camionActual, sizeof(CamionEnArchivo), 1, camiones);
```

```
    }
```

```
    fclose(camiones);
```

```
}
```

4.

```
void planificarEntregas(FILE* envios, Nodo* vCamiones[]) {
```

```
    Nodo* aux;  
    EnvioEnArchivo envioActual;  
    Paquete paqueteActual;
```

```
    fread(&envioActual, sizeof(EnvioEnArchivo), 1, envios);
```

```
    while (!feof(envios)) {
```

```
        aux = buscarEnLista (vCamiones[envioActual.idCamion-1], envioActual.cp);
```

```
        if (aux != NULL) {
```

```
            strcpy(paqueteActual.codigoPaquete, envioActual.codigoPaquete);
```

```
            strcpy(paqueteActual.direccion, envioActual.direccion);
```

```
            paqueteActual.horaEntrega = envioActual.horaEntrega;
```

```
            queue(aux->info.frente, aux->info.fin, paqueteActual);
```

```
        }
```

```
        fread(&envioActual, sizeof(EnvioEnArchivo), 1, envios);
```

```
    }
```

```
    fclose(envios);
```

```
}
```

datos al final
no se suman

5.

```
Paquete entregarPaquete (int id , int codigoP , Nodo* vCamiones[]) {
```

```
    Paquete aEntregar;
```

```
    Nodo* aux;
```

```
    aux = buscarEnLista (vCamiones[id] , codigoP);
```

```
    if (aux != NULL) {
```

```
        aEntregar = unqueue (aux->frente , aux->fin);
```

```
    }
```

```
    return aEntregar;
```

```
}
```