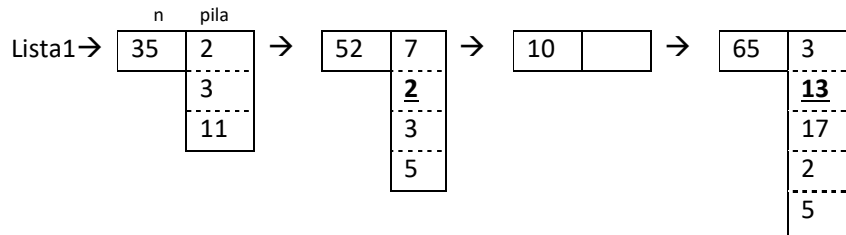
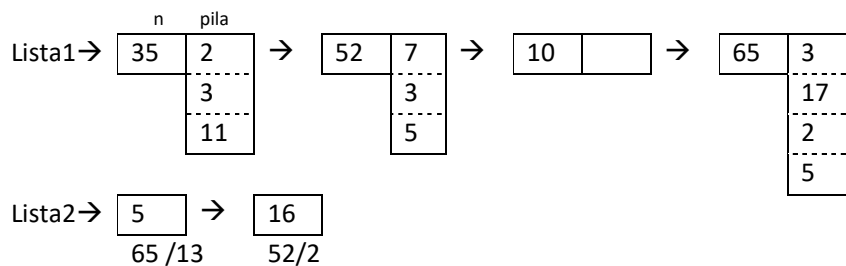


### Multiestructuras - Ejercicios adicionales

1.-Sea Lista1 una lista enlazada, cada nodo contiene un número **n** no primo y una **pila** de números primos (puede estar vacía).



Se pide recorrer Lista1 y para cada nodo evaluar si **n** es divisible por alguno de los números de la pila, en dicho caso el divisor sale de la pila y los restantes números quedan en el mismo orden, y el cociente obtenido se inserta en Lista2, una nueva lista de enteros en orden ascendente.



Desarrollar un programa en C, claro, modularizado, eficaz y eficiente que:

- cargue la Lista1 y genere la Lista2. Informar el promedio de los elementos de la lista 2
- Eliminar de la Lista2 los elementos mayores a un valor X ingresado por teclado.

La implementación del TDAPila será estática, desarrollar *ponep()* y *sacap()* para dicha implementación.

2.- Se han seleccionado un conjunto de puntos en el plano (**x,y**), y para cada uno de ellos una distancia **dist** que determina un área circular (el centro son las coordenadas **x,y**, radio **dist**), dicha información se almacena en una lista dinámica, cada nodo contiene **x, y, dist**, además una sublista de puntos (**w,z**), que distan a lo sumo **dist** de (**x,y**)

por otro lado se tiene una cola (dinámica) que almacena puntos (a,b).

Se pide sacar cada punto de la cola e insertarlo en la sublista del primer nodo, donde la distancia entre (**x,y**) y (**a,b**) sea menor o igual a **dist**.

Los puntos que no se insertaron en ninguna sublista se almacenan en una pila (estática)

Utilizar TDAPila y TDACola con las implementaciones indicadas.

3.- Desarrollar un programa que gestione las reservas de pasajes para los vuelos de una aerolínea. Para ello se cuenta con una lista de destinos con la siguiente estructura:

- Destino
- Sublista ordenada de Vuelos (a dicho destino)
  - Código de vuelo
  - Fecha y horario de salida
  - Capacidad de pasajeros
  - Sublista de pasajes vendidos (para dicho vuelo)
    - Documento del pasajero

Tanto para la reserva como para su cancelación se solicita: destino, código de vuelo y documento del pasajero. El programa debe además permitir agregar destinos y vuelos. Cada vuelo tiene una fecha y horario de salida y capacidad de pasajeros. El programa debe también listar el estado (cantidad de reservas, lugares disponibles, etc.) de todos los vuelos de un destino solicitado al operador.

NOTA: Dado que no se requieren cálculos con las fechas y horarios de salida, se sugiere almacenar dicho campo como una cadena.