Una exitosa cadena de supermercados de Quilmes nos pidió un sistema para llevar ciertas estadísticas sobre las operaciones que se hacen en cada caja de cada sucursal.
La información que tenemos consiste en:

- para cada caja, las ventas que se hicieron, cada venta como un par (importe, cantidad de artículos).
- para cada sucursal, una lista de ternas (supervisor, cajero, caja).

Esta es una muestra de información de la sucursal de Ezpeleta:

```
caja1 = [(32,8),(93,15),(140,23),(128,9),(49,7)]
caja2 = [(187,31),(428,26),(221,7),(39,2),(62,14),(194,16)]
caja3 = [(65,4),(15,7),(29,4),(94,10)]
caja4 = [(224,17),(84,9),(49,5),(58,6)]
caja5 = [(38,14),(63,10),(49,12),(21,6),(17,3),(21,2),(43,1)]
sucEzpeleta =
  [("ana","nora",caja1),("ana","juan",caja2),("ana","lucia",caja3),
   ("romina","pedro",caja4),("romina","julian",caja5)]
```

Se proveen estas funciones

```
between a b c = elem c [a..b]
esMultiploDe m n = mod n m == 0
ultimaCifra n    = mod n 10

fst3 (a,b,c) = a
snd3 (a,b,c) = b
trd3 (a,b,c) = c
```

Se pide codificar las siguientes funciones, de forma tal de usar al menos una vez cada uno de estos conceptos

- función parcialmente aplicada
- composición
- definición de lista por comprensión.
- recursividad

1. **cuantoRecaudoSupervisor**/2, recibe un supervisor y una sucursal, y devuelve el importe total recaudado por el supervisor, o sea, la suma de los importes de las operaciones de las cajas que supervisa el supervisor. P.ej. la consulta

   ```
   > cuantoRecaudoSupervisor "romina" sucEzpeleta
   ```

   devuelve `667`, que (créannos) es la suma de los importes para las cajas 4 y 5.
   Importante: armar funciones auxiliares, les van a servir en otros ítems. Les conviene leer todo el enunciado para darse cuenta cuáles.

2.
   2.a. **cantComprasDeNivel**/2, recibe un número que vamos a interpretar como un nivel, y una caja; devuelve la cantidad de compras de la caja que superan el nivel indicado. Una compra supera un nivel n si: o bien la cantidad de artículos es mayor a n, o bien el importe es mayor a 10 * n. P.ej. la consulta

   ```
   > cantComprasDeNivel 10 caja1
   ```

   devuelve `3`, porque hay tres compras de la caja uno que superan el nivel 10.
   En este también conviene definir una función auxiliar.

   2.b. **esCajaChica**/1, una caja se considera chica si ninguna de sus operaciones supera los 100 pesos.

2.c. **tieneMovimientoDeMasDe**/2, recibe importe y caja, indica si en la caja se hizo alguna operación de más importe que el indicado.

2.d. **supTuvoTrabajo**/2, recibe supervisor y sucursal, devuelve True si: ninguna de las cajas de supervisor es caja chica, y en al menos una hubo algún movimiento de más de 300 pesos. Obviamente, usar las dos anteriores.

En este punto no vale usar recursividad, en ninguno de los subpuntos.

3.

3.a. **dataSupervisor**/2, recibe supervisor y sucursal, y devuelve una terna (cajero, recaudación, cantidad de compras que superan el nivel 10) para cada caja supervisada por el supervisor. P.ej. la consulta

```
> dataSupervisor "romina" sucEzpeleta
```

devuelve `[("pedro",415,1),("julian",252,2)]`.

3.b. Dada esta definición de función

```
f x y = (x.fst.head) y > (x.snd.head) y
```

indicar su tipo, y usarla para obtener si para la primer operación de la caja 4, la última cifra del importe es mayor a la última cifra de la cantidad de artículos.

4. **comprasPorSupervisor**/1, recibe una sucursal y devuelve una lista de pares (supervisor, lista de cantidad de operaciones en las cajas de ese supervisor)**.** P.ej. la consulta

```
> comprasPorSupervisor sucEzpeleta
```

devuelve `[("ana",[5,6,4]),("romina",[4,7])]`. OJO - no se repiten los supervisores.

5.

5.a. **tieneMasComprasQueVerifican**/3, recibe una condición y dos cajas, indica si la cantidad de operaciones que cumplen el criterio en la primera es mayor que lo mismo para la segunda. La condición se evalúa para cada operación, o sea, cada par (importe, cantidad). P.ej. la consulta

```
> tieneMasComprasQueVerifican (even.snd) caja3 caja4
```

devuelve True, porque las cantidades son 3 y 1 respectivamente.

5.b. Indicar qué consulta haría para saber si la caja 3, respecto de la caja 4, tiene más:

- operaciones de más de 100 pesos
- operaciones de entre 6 y 10 artículos (ver función between)
- operaciones cuyo importe + 2 es múltiplo de 3
- cantidad de operaciones

En este subpunto no vale definir funciones auxiliares, y en todos los casos hay que usar `tieneMasComprasQueVerifican`.

6.

6.a. **esCreciente**/2, recibe un criterio y una lista, e indica si la lista es creciente (cada elemento estrictamente mayor al siguiente) según el criterio. P.ej. esta consulta

```
> esCreciente abs [1,-4,7,-14,22]
```

devuelve `True`.

6.b. Indicar qué consultas haría, usando esCreciente, para saber si son crecientes

- la cantidad de artículos de las operaciones de la caja 3
- la última cifra del importe de las operaciones de la caja 3
- el nombre de los cajeros de las cajas de la sucursal Ezpeleta
- la cantidad de compras de nivel 12 de las cajas de la sucursal Ezpeleta