Lógico 3: listas y functores

Ejercicio 1

En una competencia de saltos, cada competidor puede hacer hasta 5 saltos; a cada salto se le asigna un puntaje de 0 a 10. Se define el predicado puntajes que relaciona cada competidor con los puntajes de sus saltos, p.ej.

```
puntajes(hernan,[3,5,8,6,9]).
puntajes(julio,[9,7,3,9,10,2]).
puntajes(ruben,[3,5,3,8,3]).
puntajes(roque,[7,10,10]).
```

Se pide armar un programa Prolog que a partir de esta información permita consultar:

a. qué puntaje obtuvo un competidor en un salto, p.ej. qué puntaje obtuvo Hernán en el salto 4 (respuesta: 6).

b. si un competidor está descalificado o no. Un competidor está descalificado si hizo más de 5 saltos. En el ejemplo, Julio está descalificado.

c. si un competidor clasifica a la final. Un competidor clasifica a la final si la suma de sus puntajes es mayor o igual a 28, o si tiene dos saltos de 8 o más puntos.

Ayuda: investigar predicado **`nth1/3`** y **`nth0/3`** en el prolog.

Ejercicio 2

Se registra el ingreso de distintas personas en cada mes del año, mediante el predicado ingreso/3, p.ej.

```
ingreso(roque,enero,2000).
ingreso(roque,febrero,3500).
ingreso(roque,marzo,1200).
ingreso(luisa,enero,2500).
ingreso(luisa,febrero,850).
```

y se tiene la relación padre/2 que indica padre(Hijo, Padre), p.ej. para decir que Luisa es la madre de Roque se agrega esta cláusula

```
padre(roque,luisa).
```

(léase "uno de los padres de Roque es Luisa").

Definir estos predicados:

- **buenPasar**/1, que se verifica para una persona si se cumple alguna de estas condiciones:
  - en enero ganó más de 2200,
  - en algún mes ganó más de 3000,
  - ganó más de 2500 en dos meses distintos (esto se puede hacer sin listas, usar el \= para decir que los meses son distintos).
- **mesFilial**/2, que relaciona una persona X con un mes si en ese mes hay algún hijo de X que haya ganado más X.
  P.ej. con la información del ejemplo, la respuesta a la consulta
  ```
  ?- mesFilial(luisa,febrero).
  ```
  debería ser Yes.
  Verificar en el SWI que este predicado es totalmente inversible.
- **ingresoTotal**/2 que relaciona una persona con su ingreso total de una persona en el año; entendiendo que toda la información de la base de conocimiento corresponde al mismo año.
- **ingresoFamiliar**/2, que relaciona una persona con su ingreso familiar en el año; el ingreso familiar de una persona es su ingreso total más la suma de los ingresos totales de sus hijos.

Ejercicio 3 – subtes (orden en las listas)

En este ejercicio viene bien usar el predicado nth1/3, que relaciona un número, una lista, y el elemento de la lista en la posición indicada por el número (empezando a contar de 1).
P.ej. prueben estas consultas

- `?- nth1(X,[a,b,c,d,e],d).`
- `?- nth1(4,[a,b,c,d,e],X).`
- `?- nth1(Orden,[a,b,c,d,e],Elem).`
- `?- nth1(X,[a,b,c,d,e],j).`
- `?- nth1(22,[a,b,c,d,e],X).`

Tenemos un modelo de la red de subtes, por medio de un predicado linea que relaciona el nombre de la linea con la lista de sus estaciones, en orden. P.ej. (reduciendo las lineas)

```
linea(a,[plazaMayo,peru,lima,congreso,miserere,rioJaneiro,primeraJunta,nazca]).
linea(b,[alem,pellegrini,callao,gardel,medrano,malabia,lacroze,losIncas,urquiza]).
linea(c,[retiro,diagNorte,avMayo,independenciaC,plazaC]).
linea(d,[catedral,nueveJulio,medicina,plazaItalia,carranza,congresoTucuman]).
linea(e,[bolivar,independenciaE,pichincha,jujuy,boedo,varela,virreyes]).
linea(h,[once,venezuela,humberto1ro,inclan,caseros]).
combinacion([lima,avMayo]).
combinacion([once,miserere]).
combinacion([pellegrini,diagNorte,nueveJulio]).
combinacion([independenciaC,independenciaE]).
```

No hay dos estaciones con el mismo nombre.

Se pide armar un programa Prolog que a partir de esta información permita consultar:
a. en qué linea está una estación, predicado `estaEn/2`.
b. dadas dos estaciones de la misma línea, cuántas estaciones hay entre ellas, p.ej. entre Perú y Primera Junta hay 5 estaciones. Predicado `distancia/3` (relaciona las dos estaciones y la distancia).
c. dadas dos estaciones de distintas líneas, si están a la misma altura (o sea, las dos terceras, las dos quintas, etc.), p.ej. Independencia C y Jujuy que están las dos cuartas. Predicado `mismaAltura/2`.
d. dadas dos estaciones, si puedo llegar fácil de una a la otra, esto es, si están en la misma línea, o bien puedo llegar con una sola combinación. Predicado `viajeFacil/2`.

Ejercicio 4 – viajes

Una agencia de viajes lleva un registro con todos los vuelos que maneja de la siguiente manera:

```
vuelo(Codigo de vuelo, capacidad en toneladas, [lista de destinos] ).
```

Esta lista de destinos está compuesta de la siguiente manera:

```
escala(ciudad, tiempo de espera)
tramo(tiempo en vuelo)
```

Siempre se comienza de una ciudad (escala) y se termina en otra (no puede terminar en el aire al vuelo), con tiempo de vuelo entre medio de las ciudades. Considerar que los viajes son de ida y de vuelta por la misma ruta.

```
vuelo(ARG845, 30, [escala(rosario,0), tramo(2), escala(buenosAires,0)]).

vuelo(MH101, 95, [escala(kualaLumpur,0), tramo(9), escala(capeTown,2),
tramo(15), escala(buenosAires,0)]).

vuelo(DLH470, 60, [escala(berlin,0), tramo(9), escala(washington,2), tramo(2),
escala(nuevaYork,0)]).

vuelo(AAL1803, 250, [escala(nuevaYork,0), tramo(1), escala(washington,2),
tramo(3), escala(ottawa,3), tramo(15), escala(londres,4), tramo(1),
escala(paris,0)]).

vuelo(BLE849, 175, [escala(paris,0), tramo(2), escala(berlin,1), tramo(3),
escala(kiev,2), tramo(2), escala(moscu,4), tramo(5), escala(seul,2), tramo(3),
escala(tokyo,0)]).

vuelo(NPO556, 150, [escala(kiev,0), tramo(1), escala(moscu,3), tramo(5),
escala(nuevaDelhi,6), tramo(2), escala(hongKong,4), tramo(2),
escala(shanghai,5), tramo(3), escala(tokyo,0)]).

vuelo(DSM3450, 75, [escala(santiagoDeChile,0), tramo(1), escala(buenosAires,2),
tramo(7), escala(washington,4), tramo(15), escala(berlin,3), tramo(15),
escala(tokyo,0)]).
```

Definir los siguientes predicados; en todos vamos a identificar cada vuelo por su código.

- **tiempoTotalVuelo/2** : Relaciona un vuelo con el tiempo que lleva en total, contando las esperas en las escalas (y eventualmente en el origen y/o destino) más el tiempo de vuelo.
- **escalaAburrida/2** : Relaciona un vuelo con cada una de sus escalas aburridas; una escala es aburrida si hay que esperar mas de 3 horas.
- **ciudadesAburridas/2** : Relaciona un vuelo con la lista de ciudades de sus escalas aburridas.
- **vueloLargo/1**: Si un vuelo pasa 10 o más horas en el aire, entonces es un vuelo largo. OJO que dice "en el aire", en este punto no hay que contar las esperas en tierra.
- **conectados/2**: Relaciona 2 vuelos si tienen al menos una ciudad en común.
- **bandaDeTres/3**: Relaciona 3 vuelos si están conectados, el primero con el segundo, y el segundo con el tercero.
- **distanciaEnEscalas/3**: Relaciona dos ciudades que son escalas del mismo vuelo y la cantidad de escalas entre las mismas; si no hay escalas intermedias la distancia es 1. P.ej. París y Berlín están a distancia 1 (por el vuelo BLE849), Berlín y Seúl están a distancia 3 (por el mismo vuelo). No importa de qué vuelo, lo que tiene que pasar es que haya algún vuelo que tenga como escalas a ambas ciudades. Consejo: usar nth1.
- **vueloLento/1**: Un vuelo es lento si no es largo, y además cada escala es aburrida.