

Número:	Nome:	Turno:
---------	-------	--------

Para a resolução desta ficha apenas está permitida a seguinte utilização do módulo `lists`.

```
:- use_module(library(lists),[append/3, member/2]).
```

Se necessitar de outros predicados definidos neste módulo terá que os implementar.

Questão 1: Considere a representação de conjuntos em listas sem repetições. Defina o predicado `interseccao(+C1,+C2,-C3)` que sucede quando o conjunto `C3` é a intersecção do conjunto `C1` com o conjunto `C2`.

Questão 2: Defina o predicado `take(+N,?L1,?L2)` que sucede se a lista `L2` é o segmento inicial da lista `L1` de comprimento `N`. Se o comprimento da lista `L1` for menor do que `N`, `L2` deverá ser igual a `L1`. Se `N` for um número negativo o predicado deve falhar.

Questão 3: Defina o predicado `somas(+L,-S1,-S2)` que sucede se `S1` é o somatório dos elementos positivos da lista `L` e `S2` é o somatório dos seus elementos negativos. A implementação deste predicado só deve fazer uma travessia da lista. (Os somatórios da lista vazia deverão ser 0.)

Questão 4: Assuma que a informação referente à data de nascimento e números de telefone de um conjunto de pessoas está guardada, na base de conhecimento, em factos com o seguinte formato:

`info(nome, ano, mês, dia, lista de telefones).`

a) Defina o predicado `idades(+Ano,-Idades)` que sucede se `Idades` é a lista de pares *(nome, idade)* das pessoas no final do ano `Ano`. Só devem ser contabilizadas as pessoas já nascidas nesse ano.

b) Defina o predicado `telef(?Nome,?Telefone)` que sucede se `Telefone` é um número de telefone da pessoa `Nome`.