

```

% QUESTAO 1
% 1a. versao: remove apenas o primeiro

remover([], _, []).
remover([A|L], A, L) :- !.
remover([A|L], B, [A|L1]) :- A \= B, remover(L, B, L1).

% 2a. versao: remove todos

remover([], _, []).
remover([A|L], A, L1) :- remover(L, A, L1).
remover([A|L], B, [A|L1]) :- A \= B, remover(L, B, L1).

%=====
% QUESTAO 2
% calcsoma

calcsoma(1,1).
calcsoma(N, S) :- N1 is N - 1, calcsoma(N1,S1), S is S1 + 1 / N.

%=====
% QUESTAO 3
% usar predicado auxiliar: ligadas(A,B,L) quando A e B são ligadas
% sem passar por nenhum elemento da lista L

ligadas(A, B) :- ligadas(A, B, []).

ligadas(A, A, _) :- pessoa(A).
ligadas(A, B, L) :-
    pessoa(C),
    conhecidas(A, C),
    not(member(C, L)),
    ligadas(C, B, [A|L]).

conhecidas(A, B) :- conhece(A, B).
conhecidas(B, A) :- conhece(A, B).

% para testar

pessoa(maria).
pessoa(cibele).
pessoa(pedro).
pessoa(carlos).
pessoa(jose).

conhece(maria, pedro).
conhece(carlos, cibele).
conhece(pedro, jose).

testa :- pessoa(A), pessoa(B), writeln([A,B]),
        ligadas(A, B), writeln('sim'), fail.

```

```
testa1(X) :- ligadas(maria, X).
```

```
testa2(X, Y) :- ligadas(X, Y).
```

```
%=====
```

```
% QUESTAO 4
```

```
% arvore balanceada
```

```
balanceada(nil).
```

```
balanceada(t(L, R)) :-
```

```
    balanceada(L),
```

```
    balanceada(R),
```

```
    altura(L, AL),
```

```
    altura(R, AR),
```

```
    Diff is abs(AL - AR),
```

```
    Diff =< 1.
```

```
altura(nil, 0).
```

```
altura(t(L, R), A) :-
```

```
    altura(L, AL),
```

```
    altura(R, AR),
```

```
    A is 1 + max(AL, AR).
```

```
% para testar
```

```
arv(nil).
```

```
arv(t(L, R)) :- arv(L), arv(R).
```

```
testabal :- arv(T), writeln(T), balanceada(T), writeln('sim'), fail.
```