



Tópicos para revisão:

- Prolog: retrocesso, corte e falha
- Prolog: recursividade e listas.
- Prolog: aplicações de recursividade.
- Prolog: predicados dinâmicos.

1. [2,0] O predicado `cryptarithmic/0` mostra todas as soluções do criptograma $ODD + ODD = EVEN$. Por exemplo, uma destas soluções é $O=6, D=5, E=1, V=3$ e $N=0$, pois $655+655=1310$. Altere apenas a ordem das restrições no predicado `solution/1`, que encontra soluções por meio de retrocesso, de modo que elas sejam encontradas o mais rápido possível. Como resposta, coloque somente a nova versão do predicado `solution/1`.

```
cryptarithmic :- forall(solution(S), writeln(S)).
solution([O,D,D] + [O,D,D] = [E,V,E,N]) :-
  digit(O), digit(D), digit(E), digit(V), digit(N),
  D\=0, E\=0, E\=D, V\=0, V\=D, V\=E, N\=0, N\=D, N\=E, N\=V,
  ODD is O*100 + D*10 + D,
  EVEN is E*1000 + V*100 + E*10 + N,
  ODD+ODD == EVEN.
digit(D) :- between(0,9,D).
```

2. [2,0] Crie o predicado `all/2`, usando apenas os predicados `writeln/1` e `fail/0`, de modo que ele funcione como exemplificado ao lado (supondo que os fatos `p(1), p(2), q(3,a)` e `q(4,b)` estejam na base de conhecimentos do Prolog).

```
?- all(X, p(X)), all([X,Y], q(X,Y)).
1
2
[3,a]
[4,b]
true.
```

3. [2,0] Crie o predicado `unzip/4`, de modo que ele funcione como exemplificado ao lado (ao criar esse predicado, considere que seu primeiro argumento é um outro predicado já definido anteriormente).

```
?- unzip([X:Y,X,Y]>>true, [a:1,b:2,c:3], A, B).
A = [a, b, c],
B = [1, 2, 3].

?- unzip([Z,X,Y]>>(Z=X+Y), [1+2,3+4,5+6], A, B).
A = [1, 3, 5],
B = [2, 4, 6].
```

4. [2,0] Considerando que `nil` representa uma árvore binária vazia e que `t(X,L,R)` representa uma árvore binária com raiz `X` e subárvores `L` e `R`, crie o predicado `size(T,S)`, que calcula o tamanho `S` da árvore binária `T`.

```
?- size(t(3,t(1,nil,t(2,nil,nil)),t(4,nil,nil)), S).
S = 4.

?- size(nil, S).
S = 0.
```

5. [2,0] Crie o predicado dinâmico memoizado `f/3`, para implementar a seguinte função matemática (para um número natural $n \geq 1$):

$$f(n) = \begin{cases} 1, & \text{se } n < 4 \\ f(n-1) + f(n-2) + f(n-3), & \text{caso contrário} \end{cases}$$

```
?- time(f(40, R)).
228 inferences, 0.000 CPU in 0.000 seconds
Y = 9129195487.

?- time(f(50, R)).
% 60 inferences, 0.000 CPU in 0.000 seconds
Y = 4045078385041.
```