

**MC336 - Paradigmas de Programação**  
**Prova: Programação em Lógica - 26/10/2010**

**Questão 1** (Valor 2,5) Escreva um predicado `subset(A, B)` em Prolog que receba duas listas A e B e é satisfeito quando a lista A está contida na lista B, isto é, todos os elementos de A são membros de B. Seu predicado deve falhar em tentativas de ressatisfação, como de costume em Prolog.

*Resposta:*

```
subset([ ], _).  
subset([X|R], L) :- member(X, L), !, subset(R, L).
```

O corte é necessário para que o predicado falhe em tentativas de ressatisfação.

**Questão 2** (Valor 2,5) Considere o predicado `surpresa` definido abaixo:

```
surpresa([ ], 0).  
surpresa([_|L], N) :- surpresa(L, N1), N is N1 + 1.
```

Qual será o resultado das consultas abaixo?

```
?- surpresa([ ], N).  
?- surpresa([a,b,c], N).  
?- surpresa([a,[b,c]], N).  
?- surpresa(L, 6).  
?- surpresa([[a,b],[c,d]], 4).
```

*Resposta:*

```
?- surpresa([ ], N).  
N = 0 ;  
false  
  
?- surpresa([a,b,c], N).  
N = 3 ;  
false  
  
?- surpresa([a,[b,c]], N).  
N = 2 ;  
false  
  
?- surpresa(L, 6).  
L = [_, _, _, _, _, _] ;  
(loop infinito)  
  
?- surpresa([[a,b],[c,d]], 4).  

```

**Questão 3** (Valor 2,5) Escreva um predicado `combina(N,M,C)` que é satisfeito quando  $N$ ,  $M$  e  $C$  são números inteiros tais que  $C$  é o número de combinações de  $N$  elementos tomados  $M$  a  $M$ . O predicado deve funcionar no caso que  $N$  e  $M$  entram instanciados (isto é, com valor) e  $C$  entra não instanciada, resultando na instanciação de  $C$  com o valor correto. Seu predicado deve falhar em tentativas de resatisfação, como de costume em Prolog, e também se forem dados valores inválidos para  $N$  e  $M$ , ou seja, se  $N < 0$ ,  $M < 0$ , ou  $N < M$ .

*Resposta:*

```
combina(N,0,1) :- N >= 0.  
combina(N,M,C) :- N > 0, N1 is N-1, M > 0, M1 is M-1,  
    combina(N1,M1,C1), C is C1 * N / M.
```

As comparações  $N \geq 0$ , etc. são necessárias para garantir que o predicado falhe em argumentos inválidos.

**Questão 4** (Valor 2,5) Qual será o resultados da seguinte pergunta:

```
?- member1(E, [x, y, x, z, y]), member2(E, [z, z, x, y, y]).
```

com as seguintes definições:

```
member1(X, [X|_]).  
member1(X, [_|Y]) :- member1(X, Y).  
member2(X, [X|_]) :- !.  
member2(X, [_|Y]) :- member2(X, Y).
```

Suponha que sejam pedidas todas as resatisfações possíveis.

*Resposta:*

```
?- member1(E, [x, y, x, z, y]), member2(E, [z, z, x, y, y]).  
E = x ;  
E = y ;  
E = x ;  
E = z ;  
E = y ;  
false
```

Boa sorte a todos!