

Exercícios de Programação Lógica com Recursão em Prolog

Entrega:

Coloque os exercícios num repositório versionado, num arquivo t7.pl.

Informe a URL do repositório em <http://bit.do/entrega-paradigmas>

Prazo: segunda-feira, 28/09, 23:55

1. Explique o que está definido no predicado abaixo e quais serão as saídas das consultas. Responda esta questão em forma de comentário.

```
pred([], []).  
pred([H|T],[H1|T1]) :- H1 is H + 1, pred(T,T1)
```

```
?- pred([1,2,3],[a,b,c]).  
?- pred([8,9],L).  
?- pred([1,2,3],[2,L]).  
?- pred([1,2],[2,X]).
```

2. Defina um predicado `ziplus(L1,L2,L3)`, de forma que cada elemento da lista `L3` seja a soma dos elementos de `L1` e `L2` na mesma posição. Exemplo:

```
?- ziplus([1,2,3],[5,5,5],L).  
L = [6, 7, 8].
```

3. Defina um predicado `countdown(N,L)`, de forma que `L` seja uma lista com números de `N` a 0. Exemplo:

```
?- countdown(8,L).  
L = [8, 7, 6, 5, 4, 3, 2, 1, 0]  
?- countdown(0,L).  
L = [0]
```

4. Defina um predicado `potencias(N,L)`, de forma que `L` seja uma lista com as `N` primeiras potências de 2, sendo a primeira 2^0 e assim por diante, conforme o exemplo abaixo:

```
?- potencias(5,L).  
L = [1, 2, 4, 8, 16]  
?- potencias(0,L).  
L = []
```

Dica: defina um predicado auxiliar.

5. Defina um predicado `positivos(L1,L2)`, de forma que `L2` seja uma lista só com os elementos positivos de `L1`, conforme o exemplo abaixo:

```
?- positivos([-1,0,1,-2,9],L).  
L = [1, 9]
```

6. Considere que L1 e L2 sejam permutações de uma lista de elementos distintos, sem repetições. Sabendo disso, defina um predicado `mesmaPosicao(A,L1,L2)` para verificar se um elemento A está na mesma posição nas listas L1 e L2. Exemplo de uso:

```
?- mesmaPosicao(c,[a,b,c,d,e],[e,d,c,b,a]).
true
?- mesmaPosicao(b,[a,b,c,d,e],[e,d,c,b,a]).
false
```

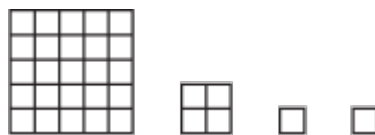
7. Crie um predicado recursivo em Prolog que receba um símbolo e uma lista, produzindo outra lista com o símbolo intercalado com os elementos da lista, conforme o exemplo a seguir:

```
?- intercala(a, [1,2,3], L).
L = [1,a,2,a,3].
?- intercala(a, [1], L.).
L = [1].
?- intercala(a, [], L).
L = [].
```

8. Dada uma lista de N alunos, deseja-se escolher NP alunos ($NP < N$) para formar uma comissão. Para isso, defina um predicado `comissao(NP,LP,C)`, que permita gerar as possíveis combinações C com NP elementos da lista LP. Exemplo:

```
?- comb(3,[maria,jose,joao,mario],C).
C = [maria, jose, joao] ;
C = [maria, jose, mario] ;
C = [maria, joao, mario] ;
C = [jose, joao, mario]
?- comb(0,[maria,jose,joao,mario],C).
C = []
```

9. (Adaptado de: OBI) Tem-se N azulejos 10cm x 10cm e, com eles, deve-se montar um conjunto de quadrados de modo a utilizar todos os azulejos dados, sem sobrepô-los. Inicialmente, deve-se montar o maior quadrado possível; então, com os azulejos que sobraram, deve-se montar o maior quadrado possível, e assim sucessivamente. Por exemplo, se forem dados 31 azulejos, o conjunto montado terá **4 quadrados**, conforme a figura abaixo:



Conjunto com quatro quadrados,
montado a partir de 31 azulejos

Para resolver este problema, você deverá definir um predicado `azulejos(NA, NQ)`, de forma que NQ seja o número de quadrados que se deve montar com NA azulejos. Dica: use os predicados `sqrt` e `floor`, pré-definidos em Prolog.