3a. Prova de IA - Listas e Functores - 2a. Parte - 28/09

IP da prova: 200.19.107.44/boca
Aquecimento individual ou no máximo dois alunos

1. Atenção aos nomes dos predicados submetidos, aridade e ordem dos argumentos. Siga exatamente os protótipos.
2. Cuidar na formatação. Esta é exatamente a dos exemplos.
3. Não há penalizações por submissões erradas.
4. Ao testarem as questões, adcionem os testes e resultados dentro de /* comentários */ no arquivo da submissão

1. Implemente um predicado que realize uma ação de um filtro no termo central, o do meio, de uma tupla-3 (functor) em uma lista de tupla-3. Veja e siga os exemplos.
Protótipo do predicado: filtro_1(L, X)

Exemplos de I/O:
?- filtro_1([(3, a, 3), (4, b, 4), (5, c, 5)], X).
X = [a, b, c].
?- filtro_1([(3, a, 3), (4, b, 4)], X).
X = [a, b].
?- filtro_1([], X).
X = [].
Grau de dificuldade: fácil

2. Implemente um predicado que construa uma lista de tuplas, a cada 3 átomos a lista. Caso o tamanho da lista não seja um múltiplo de 3, preencher como * e #, conforme os exemplos. Sobrando 2, ponha o '*' no centro, caso sobre 1 ponha '#' na primeira e terceira posição da tupla. Veja e siga os exemplos.
Protótipo do predicado: monta_1(L1,L2)

Exemplos de I/O:
X = [(a, b, c), (1, 2, 3), (4, *, 5)] ==> sobraram 2
?- monta_1([a,b,c, 1,2,3,4],X).
X = [(a, b, c), (1, 2, 3), (#, 4, #)] . ==> sobrou 1
?- monta_1([a,b,c, 1,2,3],X).
X = [(a, b, c), (1, 2, 3)] .
?- monta_1([],X).
X = []
Grau de dificuldade: fácil

3. Implemente um predicado que gere uma lista numérica de 1 a N. Onde N >= 0, ou seja, não há negativos. Veja e siga os exemplos.
Protótipo do predicado: monta_N(N,L)

Exemplos de I/O:
?- monta_N(8,X).
X = [1, 2, 3, 4, 5, 6, 7, 8].
?- monta_N(0,X).
X = [].
Grau de dificuldade: fácil

4. Implemente um predicado que realize um filtro em uma lista de tupla-2, selecionando apenas o par correspondente dos termos que X aparece em L. Ou na primeira parte ou na segunda parte. Veja os exemplos.
Protótipo do predicado: filtro_2(X , L1, L2).

Exemplos de I/O:
?- filtro_2(2, [(1,2),(b,2), (2,4),(2,x)], L).
L = [1, b, 4, x] .
?- filtro_2(2, [(1,12),(b,12), (12,4),(12,x)], L).
L = [] .
?- filtro_2(2, [], L).
L = [] .
Grau de dificuldade: fácil

5. Implemente um predicado que construa uma lista de pares. O tamanho da lista resultante, será dada pelo menor tamanho das duas listas de entrada: L1 e L2. Veja os exemplos.
Protótipo do predicado: monta_dupla(L1, L2, L3).

Exemplos de I/O:
?- monta_dupla([1,2,3,4,45],[a,b,c,d,e,f,h], X).
X = [(1, a), (2, b), (3, c), (4, d), (45, e)].
?- monta_dupla([1,2,3,4,45],[], X).
X = [].
?- monta_dupla([1,2,3,4,45],[a,b,c], X).
X = [(1, a), (2, b), (3, c)] .
Grau de dificuldade: fácil

6. Implemente um predicado que construa uma lista de lista, contendo o seu tamanho decrescente, com o respectivo número n vezes. Caso N=0, retorne uma lista vazia.
Protótipo do predicado: num_l(N, L).

Exemplos de I/O:
?- num_l(7,X).
X = [[7, 7, 7, 7, 7, 7, 7], [6, 6, 6, 6, 6, 6], [5, 5, 5, 5, 5], [4, 4, 4, 4], [3, 3, 3], [2, 2], [1]].
?- num_l(2,X).
X = [[2, 2], [1]].
Dificuldade: médio

7. Implemente um predicado que transforme várias sub-listas, em um única lista "achatada".
Protótipo do predicado: achata(X,Y)

Exemplos de I/O, vale "achata(X,Y)":
?- num_l(7,X), achata(X,Y).
X = [[7, 7, 7, 7, 7, 7, 7], [6, 6, 6, 6, 6, 6], [5, 5, 5, 5, 5], [4, 4, 4, 4], [3, 3, 3], [2, 2], [1]],
Y = [7, 7, 7, 7, 7, 7, 7, 6, 6|...].
?- num_l(3,X), achata(X,Y).
X = [[3, 3, 3], [2, 2], [1]],
Y = [3, 3, 3, 2, 2, 1].
?- num_l(0,X), achata(X,Y).
X = [],

Y = [] .
Grau de dificuldade: fácil, mas tem que pensar um pouco!

8. Implemente um predicado que transforme os termos de uma lista, em uma lista de tupla-3, com o símbolo '#' como sendo o termo do meio da tripla. Veja os exemplos.
Protótipo do predicado: duplica_x(L1,L2)

Exemplos de I/O:
?- duplica_x([3,4,5],X).
X = [(3, #, 3), (4, #, 4), (5, #, 5)].
?- duplica_x([3],X).
X = [(3, #, 3)].
?- duplica_x([],X).
X = [].
Grau de dificuldade: fácil

Boa sorte.