UFRJ faz 100 ANOS 1920 | 2020 UNIVERSIDADE FEDERAL DO RIO DE JANEIRO	CURSO: - DISCIPLINA: PROGRAMAÇÃO EM LÓGICA E IA PROFESSOR: DANIEL SCHNEIDER PERÍODO: 2022.1 DOCUMENTO: **LISTA #2 DE EXERCÍCIOS**

Q1. Considere a relação de concatenação:

```
① conc( [], L, L ).
② conc( [X|T1], L2, [X|T3] ) :- conc( T1, L2, T3 ).
```

Quais seriam as respostas do Prolog às seguintes perguntas ?
a) ?- conc([a,b,d], [f,h], L).
b) ?- conc([b,c,e], L, [b,c,e]).
c) ?- conc([b,c,e], L, [b,c,d]).
d) ?- conc(L, [b,c], [a,b,b,c]).
e) ?- conc(L1, L2, [1,2,3]).

Q2. Qual será o resultado das seguintes unificações ?

a) [X|L] = [3,4,8].

b) [X] = [].

c) [Y|T] = [[]].

d) [Y|T] = [].

e) [X|L] = [[1,2],3].

f) [Z|S] = [a, [], b].

Q3. Considere que o seguinte programa é consultado:

```
① conc( [], L, L ).
② conc( [X|T1], L2, [X|T3] ) :- conc( T1, L2, T3 ).
```

Construa uma árvore mostrando como o Prolog responderia à seguinte pergunta:

?- conc([3|T], [7,5], [X | [2,7,5]]).

Q4. Repita o exercício anterior, considerando agora a seguinte pergunta:

?- conc(T1, [3,7], [1,3,7]).

Q5. Uma lista é um *palíndromo* se ela é lida da mesma forma da esquerda para a direita e da direita para a esquerda. Por exemplo, [a,c,d,c,a], [a] e [] são exemplos de palíndromos.

a) Defina em Prolog a relação **inverte(L1, L2)**, que inverte a lista L1 obtendo a lista L2. Por exemplo, inverte([a,b,c,d], T] produziria T = [d,c,b,a].
b) Usando a relação anterior, defina o predicado **palindromo(L)**, que é verdade se L é um palíndromo.

Q6. Defina em Prolog a relação **ultimo(X, L)**, que é verdade se X é o último elemento da lista L.
Por exemplo: ultimo(a, [g,r,a]) produziria Yes, mas ultimo(a, [a,g,r]) produziria No.

Q7. Seja V1 x V2 o produto escalar de dois vetores de tamanho n. Por exemplo:

[1,3,5] x [5,2,4] = 1*5 + 3*2 + 5*4 = 31

Defina em Prolog o predicado **prodescalar(V1, V2, P)**, que é verdade se P é o produto escalar dos vetores V1 e V2. Observe que V1 e V2 devem ser listas de mesmo tamanho. Por exemplo:

?- prodescalar([1,3,5], [5,2,4], X).
X = 31

Q8. Defina em Prolog dois predicados **par(Lista)** e **impar(Lista)** de modo que eles são verdadeiros se seus argumentos são, respectivamente, listas de tamanho par e ímpar. Mostre através de uma árvore como o Prolog responderia à pergunta impar([a,b,c,d]). Exemplos:

?- par([a,b,c]).
false

?- impar([a,b,c]).
true

Q9. Defina em Prolog a relação **lshift(L1,L2)**, de forma que a lista L2 seja a lista L1 rotacionada de um elemento para a esquerda. Por exemplo, lshift([1,2,3,4], L) produziria L = [2,3,4,1].

Q10. Defina em Prolog a relação **rshift(L1,L2)**, de forma que a lista L2 seja a lista L1 rotacionada de um elemento para a *direita*. Por exemplo, rshift([1,2,3,4], L) produziria L = [4,1,2,3].

Q11. Defina em Prolog o predicado **tamanho(L, N)**, que é verdade se o número de elementos da lista L é N. Por exemplo:

?- tamanho([a,b,g,k], T).
T = 4

Q12. Defina em Prolog o predicado **nvogais(L, N)**, que é verdade se o número de vogais da lista L é N. Por exemplo:

?- nvogais([a,b,e,g,a,k], T).
T = 3

Q13. Escreva em Prolog o procedimento **split(L, P, N)**, que divide uma lista de números L em duas listas P (com os números positivos e o zero) e N (com os números negativos). Por exemplo:

?- split([2,7,-3,0,-1], P, N).
P = [2, 7, 0]
N = [-3, -1]

Q14. Defina em Prolog o predicado **removedup(L1, L2)**, que remove os elementos duplicados da lista L1, gerando uma lista L2. Por exemplo, removedup([a,b,a,b,b,c], T) produziria T = [a,b,c].

Q15. Defina em Prolog o predicado **replace(X, Y, L1, L2)** que substitui todas as ocorrências do elemento X na lista L1 pelo elemento Y, gerando uma lista L2. Por exemplo:

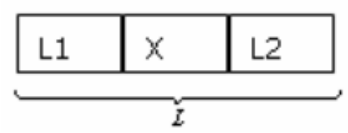
?- replace(b, c, [g,b,a,h,b,f,c], T).
T = [g,c,a,h,c,f,c]

Q16. A relação **membro(X,L)**, que define pertinência de um elemento X a uma lista L, foi definida em sala de aula da seguinte maneira:

```
① membro(X, [X|_]).
② membro(X, [Y|T]) :- membro(X,T).
```

Esta relação pode ser resumida assim:
a) X sempre pertence a uma lista encabeçada por ele mesmo.
b) X pertence a uma lista com cabeça Y e tronco T se pertence ao tronco da lista L.

É possível também definir a relação membro fazendo uso da relação de concatenação (conc). Observe a figura a seguir:



Ela nos mostra que X é membro de uma lista L se a lista L for o resultado da concatenação de alguma lista (L1) com uma lista encabeçada por X ([X|L2]). É evidente que se X for o cabeça da lista L, então L1=[]. Da mesma forma, se X for o último elemento da lista L, teremos L2=[].

Redefina a relação membro utilizando esta idéia. Uma única regra é suficiente para definir a relação.

Q17. Defina em Prolog a relação **insere2(X, L, R)**, que insere o elemento X na cabeça da lista L, obtendo uma lista resultante R, **apenas no caso em que X não pertence à lista**. Por exemplo, insere2(c, [a,b], T) produziria T=[c,a,b], mas insere2(c, [a,b,c], T) produziria T=[a,b,c].

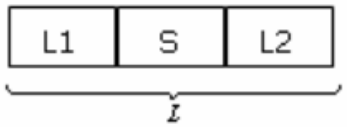
Q18. Defina em Prolog a relação **insere3(X, L, R)**, que insere o elemento X **em qualquer posição da lista L**, obtendo uma lista resultante R. Sugestão: para implementar isto, use o predicado de deleção *del/3*.

Q19. Escreva em Prolog um procedimento **permutacao(L, P)**, que gera uma permutação P da lista L. Por exemplo:

```
?- permutacao( [1,2,3], P).
P = [1, 2, 3] ;
P = [2, 1, 3] ;
P = [2, 3, 1] ;
P = [1, 3, 2] ;
P = [3, 1, 2] ;
P = [3, 2, 1] ;
No
```

Sugestão: Defina o procedimento de forma recursiva e utilize a relação *insere3/3* vista no exercício 18.

Q20. Dada uma lista L, dizemos que S é sublista de L se S estiver contida na lista L. Em outras palavras, S é sublista de L se L puder ser decomposta da seguinte forma:



Escreva um procedimento **sublista(S,L)**, que é verdade se S é sublista da lista L. Por exemplo:

```
?- sublista( [d,e,g], [b,d,e,g,h] ).
Yes

?- sublista( [d,g,h], [b,d,e,g,h] ).
No
```

Q21. Reescreva em Prolog o predicado **del(X,L1,L2)**, de forma a deletar **todas as ocorrências do elemento X** da lista L1, obtendo L2. Por exemplo:

```
?- del( b, [a,b,g,b], L ).
L = [a,g] ;
No
```

Q22. Defina em Prolog a relação **intersect(L1,L2,L3)**, que computa a intersecção das listas L1 e L2, obtendo L3. Por exemplo:

```
?- intersect( [2,4,7,8], [1,3,2,8,9,5], L ).
L=[2,8]
```

Q23. Escreva em Prolog o predicado **vizinhos/3**, para testar se dois elementos são vizinhos numa lista. Exemplos:

```
?- vizinhos(3, 8, [4,2,3,8,7]).
Yes

?- vizinhos(3, Y, [4,2,3,8,7]).
Y=2;
Y=8;
No
```

****