

## PROLOG

### Questão 1

Escreva as cláusulas para concatenar duas listas.

a) Em PROLOG: `concatenar(L1+, L2+, L3-)` .

Por exemplo:

?- `concatenar([a,b,c], [d,e,f,g,h], L)` .

Deve retornar:

`L = [a,b,c,d,e,f,g,h]` .

### Questão 2

Escreva as cláusulas para concatenar duas listas, sendo que a segunda lista vem na frente.

a) Em PROLOG: `concatenarInv(L1+, L2+, L3-)` . Neste caso dê exemplos das metas.

Por exemplo:

?- `concatenarInv([a,b,c], [d,e,f,g,h], L)` .

Deve retornar:

`L = [d,e,f,g,h,a,b,c]` .

### Questão 3

Escreva as cláusulas para concatenar uma lista de listas.

a) Em PROLOG: `concatenar(LL+, L-)` .

Por exemplo:

?- `concatenar([[a,b], [c], [e,f,g]], L)` .

Deve retornar:

`L = [a,b,c,d,e,f,g]` .

#### Questão 4

Escreva as cláusulas para juntar duas listas, intercalando seus elementos.

b) Em PROLOG: `mesclando(L1+, L2+, L3-)` .

Por exemplo:

`?- mesclando([a,b,c], [d,e,f,g,h], L)` .

Deve retornar:

`L = [a,d,b,e,c,f,g,h]` .

#### Questão 5

Escreva as cláusulas para adicionar um elemento ao final de uma lista.

a) Em PROLOG: `adicionarFinal(E+, L+, LR?)` .

Por exemplo:

`?- adicionarFinal(z, [a,b,c], L)` .

Deve retornar:

`L = [a,b,c,z]` .

#### Questão 6

Escreva as cláusulas para inverter uma lista.

a) Em PROLOG: `inverter(L+, Linv-)` .

Por exemplo:

`?- inverter([a,b,c], L)` .

Deve retornar:

`L = [c,b,a]` .

#### Questão 7

Escreva as cláusulas para inverter uma lista genérica de tal forma que todas as suas sublistas sejam também invertidas

a) Em PROLOG: `inverterLG(LG, Linv)` .

Por exemplo:

`?- inverterLG([a,b,[c,d,e]], L)` .

Deve retornar:

`L = [[e,d,c],b,a]` .

**Dica:** implemente a clausula `is_list(L)` que retorna `true` se `L` é uma lista.

### Questão 8

Escreva as cláusulas `parear` que recebe um elemento `E` e uma lista `L`, e produz a lista de pares cujo primeiro elemento é `E` e o segundo elemento é um membro de `L`.

a) Em PROLOG: `parear(E,L,LL)` .

Por exemplo:

`?- parear(x,[a,b,c], LL)` .

Deve retornar:

`LL = [[x,a],[x,b],[x,c]]` .

### Questão 9

Escreva as cláusulas `pares` que recebe uma lista `L`, e produz a lista de todos os pares não ordenados dos elementos de `L`.

b) Em PROLOG: `pares(L,LL)` .

Por exemplo:

`?- pares([a,b,c,d], LL)` .

Deve retornar:

`L = [[a,b],[a,c],[a,d],[b,c],[b,d],[c,d]]` .

### Questão 11 – Difícil.

Escreva as cláusulas para produzir todas as permutações dos elementos de uma lista.

- c) Em PROLOG: `permutar(L1,L)` . Produz uma solução para cada permuta.

Por exemplo:

```
?- permutar([1,2,3],L) .  
L = [1, 2, 3];  
L = [1, 3, 2];  
L = [2, 1, 3];  
L = [2, 3, 1];  
L = [3, 1, 2];  
L = [3, 2, 1];  
false.
```

**Dica:** use a função `remove` vista em sala:

```
remove(X,[X|T],T) .  
remove(X,[H|T],[H|NT]):- remove(X,T,NT) .
```

Sabendo que:

```
?- remove(X,[a,b,c],R) .  
X = a,  
R = [b, c];  
X = b,  
R = [a, c];  
X = c,  
R = [a, b];  
false.
```

### Questão 12

Escreva as cláusulas que recebe uma lista e testa se ela tem elementos repetidos (pode ser interpretada como um conjunto). OBS: A expressão \+ faz a negação de uma cláusula.

a) Em PROLOG: conjunto(L) .

Por exemplo:

```
?- conjunto([a,b,c,d]) .  
true.
```

```
?- conjunto([a,b,d,c,d]) .  
false.
```

### Questão 13

Escreva as cláusulas que recebe duas listas e testa se a primeira é prefixo da segunda.

a) Em PROLOG: prefixo(L1,L2) .

Por exemplo:

```
?- prefixo([a,b,c],[a,b,c,d,e,f,g]) .  
true.
```

```
?- prefixo([a,b,c],[a,b,f,g]) .  
false.
```

### Questão 14

Escreva as cláusulas que recebe duas listas e testa se a primeira é subsequência da segunda.

a) Em PROLOG: subsequencia(L1,L2) .

Por exemplo.

```
?- subsequencia([a,b,c],[d,z,a,b,c,f,g]) .  
true .
```

```
?- subsequencia([a,b,c],[d,z,a,b,f,c,g]) .  
false.
```

### Questão 15

Escreva a cláusula que recebe duas listas ordenadas e as mescla (*merge*) seus elementos em uma única lista ordenada.

- a) Em PROLOG: `mescle (L1+, L2+, L?)` .

Por exemplo.

```
?- mescle([4,5,8],[1,3,7,9,11],L) .  
L=[1,3,4,5,7,8,9,11] .
```

### Questão 16

Escreva a cláusula que recebe uma lista de números e os ordena usando o algoritmo de ordenação por mesclagem.

- a) Em PROLOG: `mergeSort (L+, LO?)` .

Por exemplo.

```
?- mergeSort([7,1,11,4,9,5,8],L) .  
L=[1,4,5,7,8,9,11] .
```

### Questão 17

Reescrevas as cláusulas de mesclagem para que possamos mesclar qualquer lista. Para isto construiremos três versões de mescle, uma cláusula que só recebe as listas, uma que recebe as listas e uma cláusula de comparação, e uma que recebe as listas, uma cláusula de comparação e uma cláusula de extração da chave de comparação dos elementos das listas:

a) `mescleV1 (L1+, L2+, L?)` . Por exemplo.

```
?- mescleV1([4,5,8],[1,3,7,9,11],L) .  
L=[1,3,4,5,7,8,9,11] .
```

b) `mescleV2 (L1+, L2+, L?, FC+)` . Por exemplo.

```
?- mescleV2([4,5,8],[1,3,7,9,11],L,=<) .  
L=[1,3,4,5,7,8,9,11] .  
?- mescleV2([8,5,4],[11,9,7,3,1],L,>) .  
L=[11,9,8,7,5,4,3,1] .
```

c) `mescleV3 (L1+, L2+, L?, FC+, Ch+)` . Por exemplo.

```
?- mescleV3([[4,2],[0,5,1,2],[8,2,3,4,2]],  
            [[2],[6,4,2],[1,3,7,9,11,2]],  
            L,=<,length) .  
  
L=[[2],[4,2],[6,4,2],[0,5,1,2],[8,2,3,4,2],[1,3,7,9,11,2]] .  
  
?- mescleV3([ [8,2,3,4,2], [4,2], [0,5,1,2] ],  
            [ [6,4,2], [2], [1,3,7,9,11,2] ],  
            L,>,first) .  
  
L=[[8,2,3,4,2],[6,4,2],[4,2],[2],[1,3,7,9,11,2],[1,5,1,2]]
```

**Dica:** use a função `call` vista em sala de aula

### Questão 18

Faça o mesmo que foi feito na Questão 17c, agora para as cláusulas `mergeSort`.

Adicione versões curtas de `mergesort`, com valores default para comparação e chave.