

Programação Funcional – Lista 1

Questão 1 (2 pontos)

Escreva as cláusulas para concatenar duas listas.

a) Em racket: `(concatenar1 l1 l2)`

Por exemplo:

```
> (concatenar1 '(a b c) '(d e f g h))
```

Deve retornar:

```
(a b c d e f g h)
```

Questão 2 (2 pontos)

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

a) Em RACKET: `(concatenarInv l1 l2)`

Por exemplo:

```
> (concatenarInv '(a b c) '(d e f g h))
```

Deve retornar:

```
(d e f g h a b c)
```

Questão 3 (3 pontos)

Escreva uma função para concatenar uma lista de listas.

a) Em RACKET: `(concatenar2 l1)`

Por exemplo:

```
> (concatenar2 '((a b) (c) (d e f g)))
```

Deve retornar:

```
(a b c d e f g)
```

Questão 4 (3 pontos)

Escreva uma função variádica para concatenar N listas.

a) Em RACKET: `(concatenar3 l ...)`

Por exemplo:

```
> (concatenar3 '(a b) '(c) '(d e f g))
```

Deve retornar:

```
(a b c d e f g)
```

Questão 5 (2 pontos)

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

b) Em Racket: `(juntar l1 l2)`

Por exemplo:

```
> (juntar '(a b c) '(d e f g h))
```

Deve retornar:

```
(a d b e c f g h)
```

Questão 6 (3 pontos)

Escreva uma função para criar uma lista intercalada de tamanho N, de dois elementos e1 e e2.

a) Em RACKET: `(intercala n e1 e2)`

Por exemplo:

```
> (intercala 5 'x 'y)
```

Deve retornar:

```
(x y x y x)
```

Questão 7 (3 pontos)

Escreva uma função para criar uma lista intercalada de tamanho N, de M elementos e_M.

a) Em RACKET: `(intercala2 n e ...)`

Por exemplo:

```
> (intercala 5 'x 'y 'z)
```

Deve retornar:

```
(x y z x y)
```

Questão 8 (3 pontos)

Escreva a função `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 RACKET: `(parear e l)`

Por exemplo:

```
> (parear 'x '(a b c))
```

Deve retornar:

```
((x a) (x b) (x c))
```

Questão 9 (3 pontos)

Escreva a função `pares` que recebe uma lista L e produz a lista de todos os pares de elementos de L.

a) Em RACKET: `(pares l)`

Por exemplo:

```
> (pares '(a b c d))
```

Deve retornar:

```
((a b) (a c) (a d) (b c) (b d) (c d))
```

Questão 10 (2 pontos)

Escreva a função que recebe uma lista e testa se ela tem elementos repetidos (pode ser interpretada como um conjunto).

a) Em RACKET: (conjunto? L)

Por exemplo:

```
> (conjunto? '(a b c d))
```

Deve retornar:

```
#t
```

```
> (conjunto? '(a b c d b))
```

Deve retornar:

```
#f
```

Questão 11 (2 pontos)

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

a) Em RACKET: (prefixo? l1 l2)

Por exemplo:

```
> (prefixo? '(a b c) '(a b c d e f g))
```

Deve retornar:

```
#t
```

```
> (prefixo? '(a b c) '(a b f g))
```

Deve retornar:

```
#f
```

Questão 12 (3 pontos)

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

a) Em RACKET: (subsequência? l1 l2)

Por exemplo.

```
> (subsequência? '(a b c) '(d z a b c f g))
```

Deve retornar:

```
#t
```

```
> (subsequência? '(a b c) '(d z a b f c g))
```

Deve retornar:

```
#f
```

Questão 13 (3 pontos)

Sem usar o comando `equal?`, escreva a função que recebe duas listas genérica e testa se elas são iguais.

a) Em RACKET: `(iguais-lg? lg1 lg2)`

Por exemplo.

```
> (iguais-lg? '(a (b c)) '(a (b c)))
```

Deve retornar:

```
#t
```

```
> (iguais-lg? '(a b c) '(d z a b f c g))
```

Deve retornar:

```
#f
```

Questão 14 (3 pontos)

Escreva a função `substitui-lg` que substitui todas as ocorrências do átomo `old` por um átomo `new` em uma estrutura de lista genérica `lg`.

```
> (substitui-lg 'c 'manoe! '(a (b c)))
```

Deve retornar:

```
'(a (b manoe!))
```

Questão 15 (3 pontos)

Escreva a função `aplanar` que recebe uma lista genérica `LG` e produz a lista plana `L` com todos os elementos de `LG`.

Por exemplo:

```
> (aplanar '(a (b c (d)) ((e))))
```

Deve retornar:

```
'(a b c d e)
```