

PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO – 2019-1  
Profa. Heloisa - Segundo Trabalho - **DATA DE ENTREGA: 19/06/2019**

=====

Escreva programas Lisp para resolver os seguintes problemas:

- 1) Dada uma lista *L* que contém elementos de qualquer tipo, possivelmente com repetições, construir outra lista que mostre quantas vezes cada elemento atômico (átomo, número, lista vazia) aparece na lista dada, inclusive nas sublistas. A lista resultante deve conter pares de elementos (pares são listas de dois elementos) sendo o primeiro elemento do par um elemento atômico que aparece na lista dada e o segundo elemento do par, o número de vezes que esse elemento aparece na lista. A ordem dos elementos na lista original deve ser mantida.

Por exemplo, dada a lista

*L* = (a b z x 4.6 (a x) () (5 z x) ())

Deve ser construída a lista

((a 2) (b 1) (z 2) (x 3) (4.6 1) ((a x) 2) (5 1) ())

(Sugestão: desparentizar a lista dada antes de fazer as outras operações.)

- 2) Dada uma lista *L* com elementos de qualquer tipo, construir outra lista em que repetições consecutivas de elementos devem ser substituídas por pares da forma (N E), onde N é o número de repetições consecutivas do elemento E.

Por exemplo, dada a lista

*L* = (a a a a b c c a a d e e e e)

Deve ser construída a lista

((4 a) (1 b) (2 c) (2 a) (1 d) (4 e))

**Observações:**

- Podem ser utilizados, se necessário, as funções pré-definidas de LISP car, cdr, cons, equal, append, listp, atom, member, length, list, not e as construções do e dolist.
- Os trabalhos podem ser feitos em duplas, que deverão ser as mesmas em todos os trabalhos;
- Usar OBRIGATORIAMENTE, CLisp para implementar o trabalho;
- Entregar, via AVA (tarefa com envio de arquivo único):
- Relatório (Arquivo txt, doc ou pdf) com listagem do código fonte, explicação do funcionamento de cada um dos procedimentos definidos e resultados de execução de pelo menos dois exemplos de cada exercício;
- Arquivo do LISP com o código fonte do trabalho.
- **DATA DE ENTREGA: 19/06/2019**