

# T2 (PLP - 2019)

Autores:

Alexandre Regali Seleglim RA: 551473

Pedro de Oliveira Zani RA: 408565

O programa possui 7 funções:

`desparentiza`(L) [usado em ambos os problemas]

`contaElemento`(E L) [usado no primeiro problema]

`elimina`(E L) [usado no primeiro problema]

`montaPares`(L) [usado no primeiro problema]

`contaConsec`(E L) [usado no segundo problema]

`removeConsec`(L) [usado no segundo problema]

`montaParesConsec`(L) [usado no segundo problema]

## DESPARENTIZA:

Desparentiza a lista de entrada removendo todas as sublistas e adicionando seus elementos a lista principal.

Exemplo:

```
(desparentiza '(a b z x 4.6 (a x) () (5 z x) ()))
```

Retorna: (A B Z X 4.6 A X NIL 5 Z X NIL)

## CONTAELEMENTO:

Conta todas as ocorrências de um elemento (primeiro argumento da função) em uma dada lista (segundo argumento de uma função).

Exemplo:

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

Retorna: 3

## ELIMINA:

Remove todas as ocorrências de um elemento (primeiro argumento da função) em uma dada lista (segundo argumento de uma função).

Exemplo:

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

Retorna: (B C D E E J)

## MONTAPARES:

Monta uma lista contendo a contagem de quantas ocorrências de cada elemento há numa dada lista (argumento da função)

Exemplo:

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

Retorna: ((A 3) (B 1) (C 1) (D 1) (E 2) (J 1))

### CONTA CONSEC:

Conta todas as ocorrências CONSECUTIVAS de um elemento (primeiro argumento da função) em uma dada lista (segundo argumento de uma função).

Exemplo:

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

Retorna: 3

### REMOVE CONSEC:

Remove todas as ocorrências CONSECUTIVAS de um elemento (primeiro argumento da função) em uma dada lista (segundo argumento de uma função).

Exemplo:

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

Retorna: (B C D A E A E J)

### MONTA PARES CONSEC:

Monta uma lista contendo a contagem de quantas ocorrências de cada elemento há numa dada lista (argumento da função)

Exemplo:

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

Retorna: ((A 3) (B 1) (C 1) (D 1) (A 1) (E 1) (A 1) (E 1) (J 1))

Exemplos:

**Result**

**\$clisp main.lisp**

Exercicio 1 utilizando a lista L = (a b z x 4.6 (a x) () (5 z x) ()):

((A 2) (B 1) (Z 2) (X 3) (4.6 1) (NIL 2) (5 1))

Exercicio 2 utilizando a lista L = (a a a a b c c a a d e e e e):

((4 A) (1 B) (2 C) (2 A) (1 D) (4 E))

**Result**

**\$clisp main.lisp**

Exercicio 1 utilizando a lista L = (a b (j k) (y x (m m j) 2) () g (()) 1 2):

((A 1) (B 1) (J 2) (K 1) (Y 1) (X 1) (M 2) (2 2) (NIL 2) (G 1) (1 1))

Exercicio 2 utilizando a lista L = (a a (j k j j) (y (m m j) ) () g (()) 1 1 2):

((2 A) (1 J) (1 K) (2 J) (1 Y) (2 M) (1 J) (1 NIL) (1 G) (1 NIL) (2 1) (1 2))

Código-fonte:

```
;; funcao desparentiza (usada nos dois exercicios)

(defun desparentiza(L)
  (cond ((null L) ())
        ((atom (car L)) (append (list (car L)) (desparentiza
(cdr L)))))
        (t (append (desparentiza(car L)) (desparentiza(cdr L)))))
  )
)

;; funcoes do primeiro exercicio

(defun contaElemento(E L)
  (cond ((null L) 0)
        ((equal E (car L)) (+ 1 (contaElemento E (cdr L))))
        (t (contaElemento E (cdr L))))
  )
)

(defun elimina(E L)
  (cond ((null L) ())
        ((equal E (car L)) (elimina E (cdr L)))
        (t (append (list (car L)) (elimina E (cdr L)))))
  )
)

(defun montaPares(L)
  (cond ((null L) ())
        (t (append (list (append (list (car L)) (list (contaElemento
(car L) L))))) (montaPares (elimina (car L) L)))))
  )
)
```

```
;; funcoes do segundo exercicio
```

```
(defun contaConsec(E L)
  (cond ((null L) 0)
        ((equal E (car(cdr L))) (+ 1 (contaConsec E (cdr L))))
        ((not(equal E (car(cdr L)))) 1)
  )
)
```

```
(defun removeConsec(L)
  (cond ((null L) ())
        ((equal (car L) (car(cdr L))) (removeConsec (cdr L)))
        (t (cdr L))
  )
)
```

```
(defun montaParesConsec(L)
  (cond ((null L) ())
        (t (append (list (append (list (contaConsec (car L) L))
                                     (list (car L))))
                     (montaParesConsec (removeConsec L))))
  )
)
```

```
;; chamadas para executar o programa
```

```
(write-line "Exercicio 1 utilizando a lista L = (a b z x 4.6 (a x) ()
(5 z x) ()):")
(print
  (montaPares (desparentiza '(a b z x 4.6 (a x) () (5 z x) ())))
)
```

```
(write-line "")
(write-line "")
(write-line "")
(write-line "")
```

```
(write-line "Exercicio 2 utilizando a lista L = (a a a a b c c a a d e
e e e):")
(print
  (montaParesConsec (desparentiza '(a a a a b c c a a d e e e e)))
)
```