

# T1 (PLP - 2019)

Autores:

Alexandre Regali Seleglim RA: 551473

Pedro de Oliveira Zani RA: 408565

O programa possui 9 predicados:

**desparentize**(Lista\_in, Lista\_out),  
**tira\_comuns**(Lista1\_in, Lista2\_in, Lista\_out),  
**conta**(Elemento, Lista\_in, Numero\_ocorrencias),  
**remove\_elemento**(Elemento, Lista\_in, Lista\_out),  
**remove\_variaveis**(L1, Lout),  
**tamanho**(L1, tamanho),  
**ehVazia**(L1),  
**final**(Lista\_in, Lista\_out),  
**monta\_pares**(Lista1\_in, Lista2\_in, Lista\_out).

## **DESPARENTIZE:**

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

Exemplo:

`desparentize([abacaxi, [cebola, maçã, [morango]], [[[manga]]]], Lout)`  
`Lout = [abacaxi, cebola, maçã, morango, manga]`

## **TIRA\_COMUNS:**

Tira todos os objetos da Lista 1 que pertencem a Lista 2.

Exemplo:

`tira_comuns([a, b, c, d], [a, j, c, k], Lout)`  
`Lout = [b, d]`

## **CONTA:**

Conta todas as ocorrências de um certo elemento em uma lista.

Exemplo:

`conta(juninho, [felipe, ana, juninho, pedro, juninho, lucas, ana], N)`  
`N = 2`

## **REMOVE\_ELEMENTO:**

Remove todas as ocorrências de um certo elemento em uma lista.

Exemplo:

`remove_elemento(juninho, [felipe, ana, juninho, pedro, juninho, lucas, ana], Lout)`  
`Lout = [felipe, ana, pedro, lucas, ana]`

**REMOVE\_VARIAVEIS:**

Remove todas as ocorrências de variáveis em uma lista.

Exemplo:

```
remove_variaveis([a, b, F, c, Z], Lout)
```

```
Lout = [a, b, c]
```

**TAMANHO:**

Retorna o tamanho de uma lista.

Exemplo:

```
tamanho([a, b, c, d], N)
```

```
N = 4
```

**EHVAZIA:**

Retorna verdadeiro se a lista for vazia.

Exemplo:

```
ehVazia([])
```

```
true
```

**FINAL:**

Cria uma lista contendo sublistas com o par de cada elemento de uma outra lista com o seu número de ocorrências. Os seguintes passos são seguidos:

- Conta o primeiro elemento da Lista\_in
- Adiciona o par [Elemento, N] à Lista\_out
- Remove todas as ocorrências de Elemento da Lista\_in
- Repete recursivamente até que Lista\_in fique vazia
- Imprime a Lista\_out (write(Lout))

Exemplo:

```
final([mia, juninho, batman, mia, mia, batman], Lout)
```

```
Lout = [[mia, 3], [juninho, 1], [batman, 2]]
```

**MONTA\_PARES:**

Prepara as listas de entrada para depois serem usadas no predicado FINAL. Ele segue os seguintes passos:

- Desparentiza Lista 1
- Desparentiza Lista 2
- Remove as variáveis da Lista 1
- Remove as variáveis da Lista 2
- Tira não comuns da Lista 1 em relação à Lista 2
- Tira não comuns da Lista 2 em relação à Lista 1
- Junta as duas listas criando Lista 3 (append)
- Chama o predicado FINAL com a Lista 3 O programa principal está lendo Lista 1 e Lista 2 e depois chamará monta\_pares(Lista 1, Lista 2, Lista\_out).

## Exemplos:

```
SWI-Prolog -- c:/Users/xand/Desktop/t1.pl
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- programa.
Digite a lista 1: [a, b, Z, [a, x], [5,x], [x], par(c,d)].

Digite a lista 2: |: [4.6, 5, w, [], F, b, [], par(c,d), [], par(1,2)].

[[a,2],[x,3],[4.6,1],[w,1],[[],3],[par(1,2),1]]
true .

?-
```

```
SWI-Prolog -- c:/Users/xand/Desktop/t1.pl
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- programa.
Digite a lista 1: [a, b, [d], [j, y], c].

Digite a lista 2: |: [j, k, [[c]], b, j, a, j].

[[d,1],[y,1],[k,1]]
true .

?-
```

## Código-fonte:

```
%T1
%Autores:
%Alexandre Regali Selegim (RA 551473)
%Pedro de Oliveira Zani (RA 408565)

programa:-
write('Digite a lista 1: '),
read(Lista1),
nl,
write('Digite a lista 2: '),
read(Lista2),
nl,
monta_pares(Lista1,Listas2,_).

%listas do exemplo1:
%[a, b, Z, [a, x], [5,x], [x], par(c,d)].
%[4.6, 5, w, [], F, b, [], par(c,d), [], par(1,2)].
```

```

%%listas do exemplo2:
%[a, b, [d], [j, y], c].
%[j, k, [[c]], b, j, a, j].

%funcao desparentize
%L1, Lout
desparentize([], []).
desparentize([X|Y], [X|Z]) :- not(is_list(X)), desparentize(Y,Z),
!.
desparentize([X|Y], [X|Z]) :- is_list(X), ehVazia(X),
desparentize(Y,Z), !.
desparentize([X|Y], Z) :- is_list(X), not(ehVazia(X)),
desparentize(X,X1), desparentize(Y,Y1), append(X1, Y1, Z), !.

%funcao de tirar, de L1, os elementos comuns entre L1 e L2
%L1, L2, Lout
tira_comuns([], _, []).
tira_comuns([X1|Y1], L2, [X1|Z]) :- not(member(X1, L2)),
tira_comuns(Y1, L2, Z).
tira_comuns([_|Y1], L2, Z) :- tira_comuns(Y1, L2, Z).

%funcao de contar as ocorrencias de um elemento numa dada lista L1
%Elemento, L1, N
conta(_, [], 0).
conta(X, [X | T], N) :- !, conta(X, T, N1), N is N1 + 1.
conta(X, [_ | T], N) :- conta(X, T, N).

%funcao de remover todas as ocorrencias de um elemento numa lista
L1
%Elemento, L1, Lout
remove_elemento(_, [], []).
remove_elemento(X, [X|Y], Z) :- !, remove_elemento(X, Y, Z).
remove_elemento(X, [T|Y], Z) :- !, remove_elemento(X, Y, Y2),
append([T], Y2, Z).

%funcao de remover todas as variaveis de uma lista L1
%L1, Lout
remove_variaveis([], []).
remove_variaveis([X|Y], [X|Z]) :- not(var(X)),
remove_variaveis(Y,Z), !.
remove_variaveis([X|Y], Z) :- var(X), remove_variaveis(Y,Z), !.

%calcula o tamanho da lista L1

```

```

%L1
tamanho([],0).
tamanho([_|T],N) :- tamanho(T,X), N is X+1.

%verifica se a lista L1 eh vazia
%L1
ehVazia(L) :- tamanho(L,X),X==0.

%funcao que constroi a lista final com o par [elemento, numero de
ocorrencias] da lista L1
%L1, Lout
final([], Lout) :- length([], J), J == 0, write(Lout), !.
final([X|Y], Lout) :-
conta(X,[X|Y],N),
append(Lout,[X,N],Laux),
remove_elemento(X, [X|Y], Lout2),
final(Lout2, Laux).

%funcao que trata as duas listas e depois chama a funcao final
monta_pares(A,B,Lout) :-
desparentize(A, A2),
desparentize(B, B2),
remove_variaveis(A2, A3),
remove_variaveis(B2, B3),
tira_comuns(A3,B3, A4),
tira_comuns(B3,A3, B4),
append(A4,B4, L3),
final(L3, Lout).

```