listprocessing.pro Demo

Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.2)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

?- consult('listprocessi
true.

?- writelist([]).
true.

?- writelist([1,2,3]).
1
2
3
true.

?- member(2,[1,2,3]).
true .

?- member(4,[1,2,3]).
false.

?- count([9,8,7,6,5,4,3,2,1,0],X).
X = 10.

?- count([],X).
X = 0.

?- item(6,[1,2,3,4,9,0,8,3,2],X).
X = 8 .

?- item(6,[1,2,3],X).
false.

```
?- item(0,[1,2,3],X).
X = 1 .

?- append([1,2,3],[4,5],X).
X = [1, 2, 3, 4, 5].

?- last([4,3,5,1,3,4,5],X).
X = 5 .

?- last([1],X).
X = 1 .

?- remove(1,[4,2,1,9,1,4,0],X).
X = [4, 2, 9, 1, 4, 0] .

?- remove(3,[1,2,7,3],X).
X = [1, 2, 7] .

?- replace(2,3,[1,2,7,3,9],X).
X = [1, 2, 3, 3, 9] .

?- replace(3,4,[1,2,3,3,9],X).
X = [1, 2, 3, 4, 9].

?- replace(4,5,[1,2,3,4,9],X).
X = [1, 2, 3, 4, 5] .

?- makelist(6,2,X).
X = [2, 2, 2, 2, 2, 2] .

?- reverse([1,2,3,4,5],X).
X = [5, 4, 3, 2, 1] .

?- lastput(6,[1,2,3,4,5],X).
X = [1, 2, 3, 4, 5, 6] .

?- pick([1,2,3,4,5,6,7,8,9],X).
X = 9 .
```

```
?- pick([1,2,3,4,5,6,7,8,9],X).
X = 7 .

?- pick([1,2,3,4,5,6,7,8,9],X).
X = 9 .

?- pick([1,2,3,4,5,6,7,8,9],X).
X = 7 .

?- pick([1,2,3,4,5,6,7,8,9],X).
X = 8 .

?- pick([1,2,3,4,5,6,7,8,9],X).
X = 2 .

?- take([2,3,5,9,0],X,List).
X = 3,
List = [2, 5, 9, 0] .

?- take([2,3,5,9,0],X,List).
X = 3,
List = [2, 5, 9, 0] .

?- take([2,3,5,9,0],X,List).
X = 3,
List = [2, 5, 9, 0] .

?- take([2,3,5,9,0],X,List).
X = 9,
List = [2, 3, 5, 0] .

?- iota(12,List).
List = [1, 2, 3, 4, 5, 6, 7, 8, 9|...] .

?- iota(5,List).
List = [1, 2, 3, 4, 5] .

?- sum(iota(5,X),Y).
```

false.

```
?- sum([5,5],X).
X = 10.

?- sum([2,4,5,7,7,5,2,0,7],X).
X = 39.

?- min([2,4,5,7,7,5,2,0,7],X).
X = 0.

?- min([9,8,6,9,8,7],X).
X = 6.

?- max([9,8,6,9,8,7],X).
X = 9.

?- max([1,1,0,1,0,1,0,0,0,1,0],X).
X = 1.

?- sort_inc([5,4,9,3,8,5,4,3],X).
X = [3, 3, 4, 4, 5, 5, 8, 9] .

?- sort_inc([8,3,9,2,4,0,5,7,1,6],X).
X = [0, 1, 2, 3, 4, 5, 6, 7, 8|...] .

?- alist([4,7,8,2],[5,7,6,4],X).
X = [[4, 5], [7, 7], [8, 6], [2, 4]].

?- assoc([[a,b],[1,2]],1,Value).
Value = 1.

?- rssoc([[a,b],[1,2]],2,Value).
Value = 2.

?- flatten([a,b,c,[],[d],[a,a]],X).
X = [a, b, c, [], [[d], [[a|...], []]]] .

?- halt.
```