=====LIST PROCCESSING DEMO ==============

Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.3)

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('listStuff.pro').

true.

?- writelist([red,blue,green]).

red

blue

green

true.

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

true .

?- item(2,[red,green,blue,yellow],X).

X = blue .

?- append([oh,my],[golly,gosh],NL).

NL = [oh, my, golly, gosh].

?- last([uno,dos,tri],X).

X = tri .

?- last([butNoLeast],X).

X = butNoLeast .

?- remove(others,[one,of,these,things,is,not,like,the,others],X).

X = [one, of, these, things, is, not, like, the] .

?- replace(0,winner,[dork,pro,champ,hero],X).

X = [winner, pro, champ, hero] .

?- makelist(99, redballons,X).

X = [redballons, redballons, redballons, redballons, redballons, redballons, redballons, redballons, redballons|...] .

?- reverse([4,5,6],X).

X = [6, 5, 4] .

?- reverse([joker,queen,king,ace],X).

X = [ace, king, queen, joker] .

?- lastput(me,[friend,randomguy,badperson,slacker],X).

X = [friend, randomguy, badperson, slacker, me] .

?- pick([red,orange,green,blue,indigo,violet],Item).

Item = green .

?- pick([red,orange,green,blue,indigo,violet],Item).

Item = blue .

?- pick([red,orange,green,blue,indigo,violet],Item).

Item = violet .

?- take([red,orange,green,blue,indigo,violet],Element,Rest).

Element = orange,

Rest = [red, green, blue, indigo, violet] .

?- take([red,orange,green,blue,indigo,violet],Element,Rest).

Element = green,

Rest = [red, orange, blue, indigo, violet] .

?- iota(7,IotaN).

IotaN = [1, 2, 3, 4, 5, 6, 7] .

?- iota(10,IotaN).

IotaN = [1, 2, 3, 4, 5, 6, 7, 8, 9|...] .

?- sum([2,4,6,8],Sum).

Sum = 20.

?- min([2,4,6,8],Sum).

Sum = 2.

?- max([2,4,6,8],Sum).

Sum = 8.

?- sort_inc([5,47,69,17,25],S).

S = [5, 17, 25, 47, 69] .

?- sort_dec([5,47,69,17,25],S).

S = [69, 47, 25, 17, 5] .

?- alist([red,hot,cold],[car,pizza,drink],X).

X = [pair(red, car), pair(hot, pizza), pair(cold, drink)].

?- assoc([pair(red,car),pair(hot,pizza),pair(cold,drink)],red,X).

X = car .

?- flatten([[a,b],[c,d]],FL).

FL = [a, b, c, d] .