listprocessing.pro

```
% FILE: listprocessing.pro
% TYPE: Prolog Source
% Line: List Processing program
% DATE: December 8, 2015

writelist([]).
writelist([H|T]):-
write(H),nl,writelist(T).

member(X,[X|_]). member(X,[_|Y]) :-
member(X,Y).

count([],0).
count([_|T], L) :-
count(T, K),
L is (1 + K).

item(N, [H|_], H) :-
N = 0.
item(N,[_|T], E) :-
N > 0,
K is N-1,
item(K,T,E).

append([],L,L).
append([H|T1], L2, [H|T3]) :-
append(T1,L2, T3).
append(L1, L2,L3, Result) :-
append(L1,L2,L12),
append(L12, L3, Result).
append(L1,L2,L3,L4,Result):-
append(L1,L2,L3,L123),
append(L123,L4,Result).

last([H|[]],H).
last([_|T], Result) :-
last(T,Result).
```

```
remove(_,[],[]).
remove(First, [First|Rest], Rest).
remove(Element, [First|Rest], [First|RestLessElement]) :-
remove(Element, Rest, RestLessElement).

replace(0, Object,[_|T], [Object|T]).
replace(ListPosition, Object, [H|T1], [H|T2]):-
K is ListPosition -1,
replace(K,Object,T1,T2).

makelist(0,_,[]). makelist(Length,Element,[Element|Rest]):-
K is Length -1,
makelist(K,Element,Rest).

reverse([],[]).
reverse([H|T],R):-
reverse(T,Rev),
lastput(H,Rev,R).
lastput(E,[],[E]).

lastput(E,[H|T], [H|L]) :-
lastput(E,T,L).

pick(L, Item):-
length(L, Length),
random(0,Length,RN),
item(RN,L,Item).

take(List, Element,Rest):-
pick(List,Element), remove(Element, List, Rest).

iota(0,[]). iota(N,IotaN):-
K is N -1,
iota(K,IotaK),
lastput(N,IotaK,IotaN).

sum([],0). sum([Head|Tail],Sum):-
sum(Tail,SumOfTail),
Sum is Head + SumOfTail.
```

```
min([X], X):- !.
min([X,Y|T], N):-
(X > Y->
        min([Y|T],N)
        ;
        min([X|T],N)).
max([X], X):- !.
max([X,Y|T], N):-
(X > Y ->
        max([X|T], N)
        ;
        max([Y|T], N)).

sort_dec(L1, L2) :-
sort(L1, Tmp),
reverse(Tmp, L2).

sortA(List,Result):-
sort(List,Result).

sort_inc(List,Sorted):-
b_sort(List,[],Sorted).

b_sort([],Acc,Acc).
b_sort([H|T],Acc,Sorted):-
bubble(H,T,NT,Max),
b_sort(NT,[Max|Acc],Sorted).

bubble(X,[],[],X).
bubble(X,[Y|T],[Y|NT],Max):-
X>Y,
bubble(X,T,NT,Max).

bubble(X,[Y|T],[X|NT],Max):-
X=<Y,
bubble(Y,T,NT,Max).

alist([X], [Y], [[X,Y]]):-!.
```

```
alist([], [], []).

alist([X|L1], [Y|L2], [[X,Y]|L3]):-
alist(L1, L2, L3).

assoc([[X,_]|Tail],Key,Value):- (Key = X ->
Value = Key;
assoc(Tail,Key,Value)).

rssoc([[_,X]|Tail],Key,Value):- (Key = X ->
Value = Key;
rssoc(Tail,Key,Value)).

flatten([], []).
flatten([H|T], L) :-
atom(H),
flatten(T, Tflattened),
L= [H|Tflattened].

flatten([H|T],L) :-
flatten(H, FlatHead),
flatten(T,FlatTail),
L= [FlatHead, FlatTail].
```