

luisestevez@Luis-MacBook-Air Assignments % swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.3.6)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- module(demo), consult('lp.pro').
Warning: demo is not a current module (created)
true.

demo: ?- write_list([a,b,c,d]).
a
b
c
d
true.

demo: ?- write_list_reversed([a,b,c,d]).
d
c
b
a
true.

demo: ?- first([a,b,c,d], First).
First = a.

demo: ?- rest([a,b,c,d], Rest).
Rest = [b, c, d].

demo: ?- last([a,b,c,d], Last).
Last = d .

demo: ?- nth(2, [a,b,c,d,e], NthElement).
NthElement = c .

demo: ?- item(3, [a,b,c,d,e,f], Item).
Item = d .

demo: ?- member(a, [a,b,c,d]).
true .

demo: ?- count(a, [a,b,b,c,d], Count).
Count = 1 .

demo: ?- count(b, [a,b,b,c,d], Count).
Count = 2 .

demo: ?- append([a,b],[c,d], Result).
Result = [a, b, c, d].

demo: ?- remove(4, [1,2,3,4,5], NewList).
NewList = [1, 2, 3, 5] .

demo: ?- replace(2, h, [a,b,c,d,e], NewList).
NewList = [a, b, h, d, e] .

demo: ?- make_list(5, e, NewList).
NewList = [e, e, e, e, e] .

demo: ?- reverse([1,2,3,4,5], ReversedList).
ReversedList = [5, 4, 3, 2, 1] .

demo: ?- add_first(e, [a,b,c,d], NewList).
NewList = [e, a, b, c, d].

demo: ?- pick([1,2,3,4,e,5], Item).
Item = 5 .

demo: ?- pick([1,2,3,4,e,5], Item).
Item = e .

demo: ?- add_last(e, [a,b,c,d], NewList).
NewList = [a, b, c, d, e] .

demo: ?- take([a,b,c,d], b, NewList).
NewList = [a, c, d] .

demo: ?- iota(0, IotaK).
IotaK = [] .

demo: ?- iota(1, IotaK).
IotaK = [1] .

demo: ?- iota(6, IotaK).
IotaK = [1, 2, 3, 4, 5, 6] .

demo: ?- sum([1,2,4,5],Sum).
Sum = 12.

demo: ?- sum([1,2,4,5,6],Sum).
Sum = 18.

demo: ?- product([3,2,5],Product).
Product = 30.

demo: ?- factorial(2, F).
F = 2 .

demo: ?- factorial(5, F).
F = 120 .

demo: ?- factorial(5, 120).
true .

demo: ?- min([5,3,2,4,6], Minimum).
Minimum = 2.

demo: ?- min([5,30,7,4,6], Minimum).
Minimum = 4.

demo: ?- max([5,30,7,4,6], Maximum).
Maximum = 30.

demo: ?- sort_inc([5,3,17,21,1], Sorted).
Sorted = [1, 3, 5, 17, 21] .

demo: ?- sort_dec([5,3,17,21,1], Sorted).
Sorted = [21, 17, 5, 3, 1] .

demo: ?- alist([1,2,3],[a,b,c], AList).
AList = [pair(1, a), pair(2, b), pair(3, c)].

demo: ?- alist([1,2],[a,b,c], AList).
false.

demo: ?- assoc([pair(a,3), pair(b,5), pair(c,7)], b, Value).
Value = 5.

demo: ?- assoc([pair(a,3), pair(b,5), pair(c,7)], c, Value).
Value = 7.

demo: ?- freq_count([a,a,a,b,b,b,c,c,d], FCList).
FCList = [pair(a, 4), pair(b, 3), pair(c, 2), pair(d, 1)] .

demo: ?- freq_count([a,b,a,a,b,b,a,c,c,d], FCList).
FCList = [pair(b, 3), pair(a, 4), pair(c, 2), pair(d, 1)] .

demo: ?- freq_count([a,a,a,a,b,b,b,c,c,d], FCList).
FCList = [pair(a, 4), pair(b, 3), pair(c, 2), pair(d, 1)] .

demo: ?- freq_count([a,b,a,a,b,b,a,c,c,d], FCList).
FCList = [pair(b, 3), pair(a, 4), pair(c, 2), pair(d, 1)] .

demo: ?- make_set([1,2,3], Set).
Set = [1, 2, 3].

demo: ?- make_set([1,a,3], Set).
Set = [1, a, 3].

demo: ?- make_set([1,a,3,3], Set).
Set = [1, a, 3] .

demo: ?- make_set([1,a,3,3,a,a], Set).
Set = [1, 3, a] .

demo: ?- flatten([a,[b],c,d,[e,f],g], FL).
FL = [a, b, c, d, e, f, g] .

demo: ?- halt.

luisestevez@Luiss-MacBook-Air Assignments %