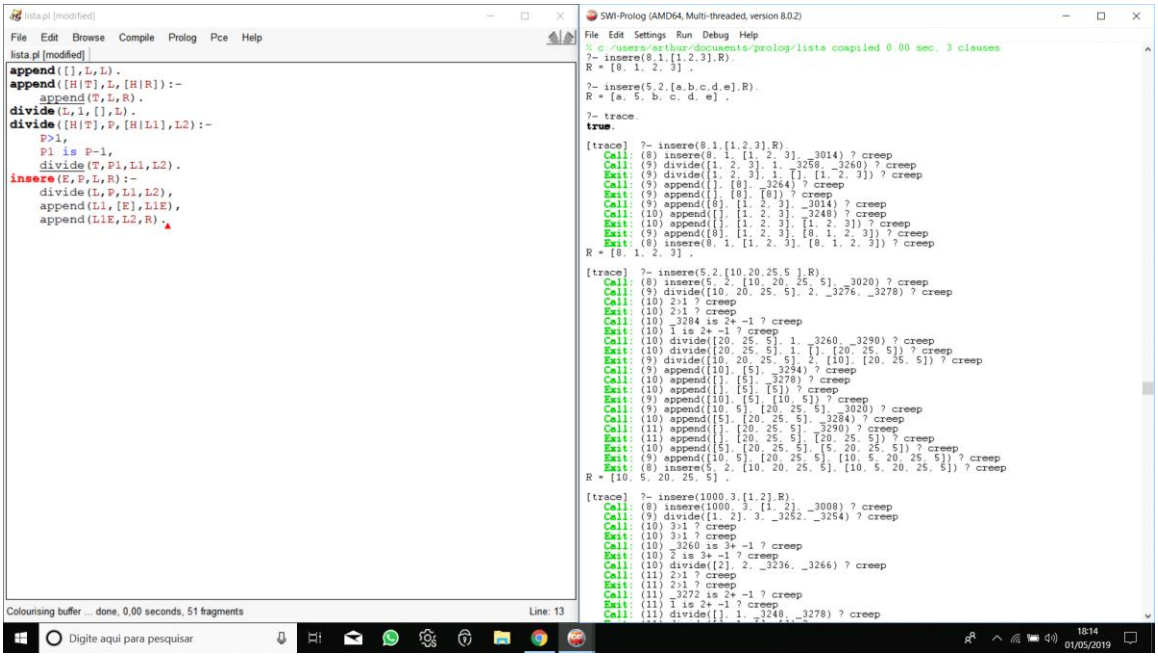
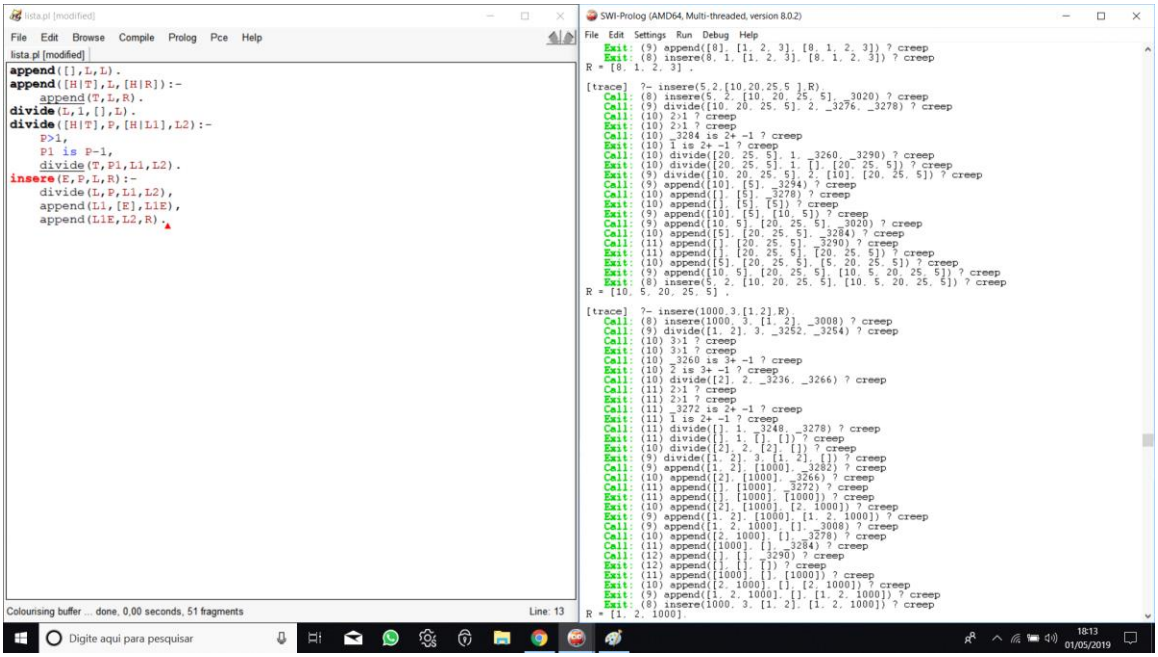
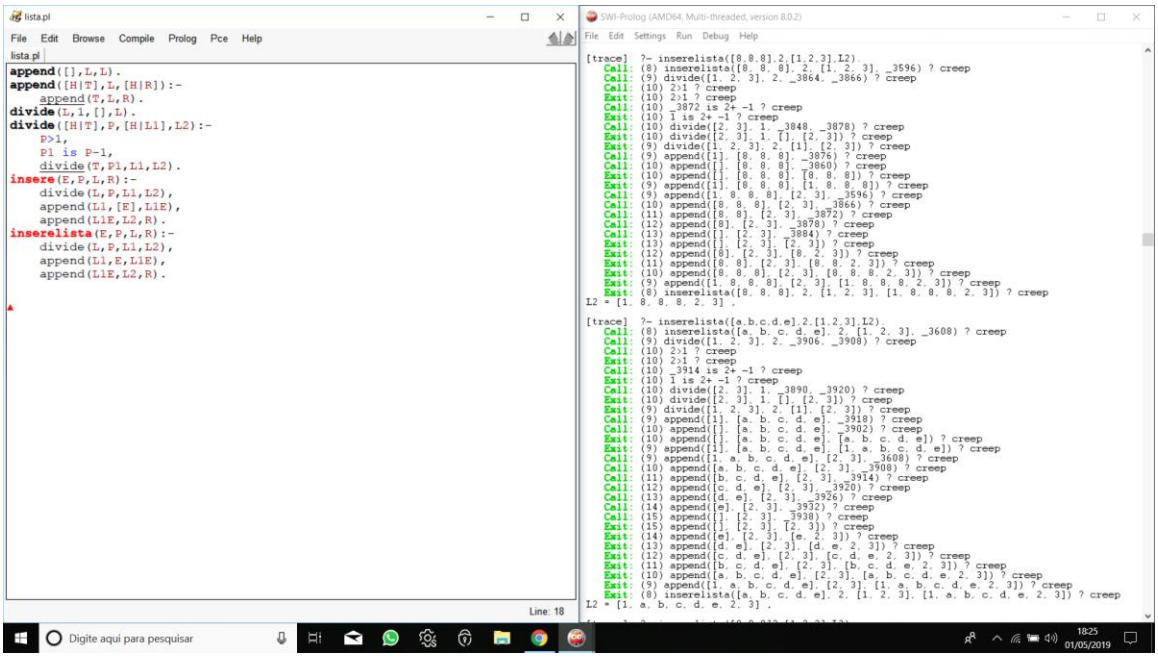
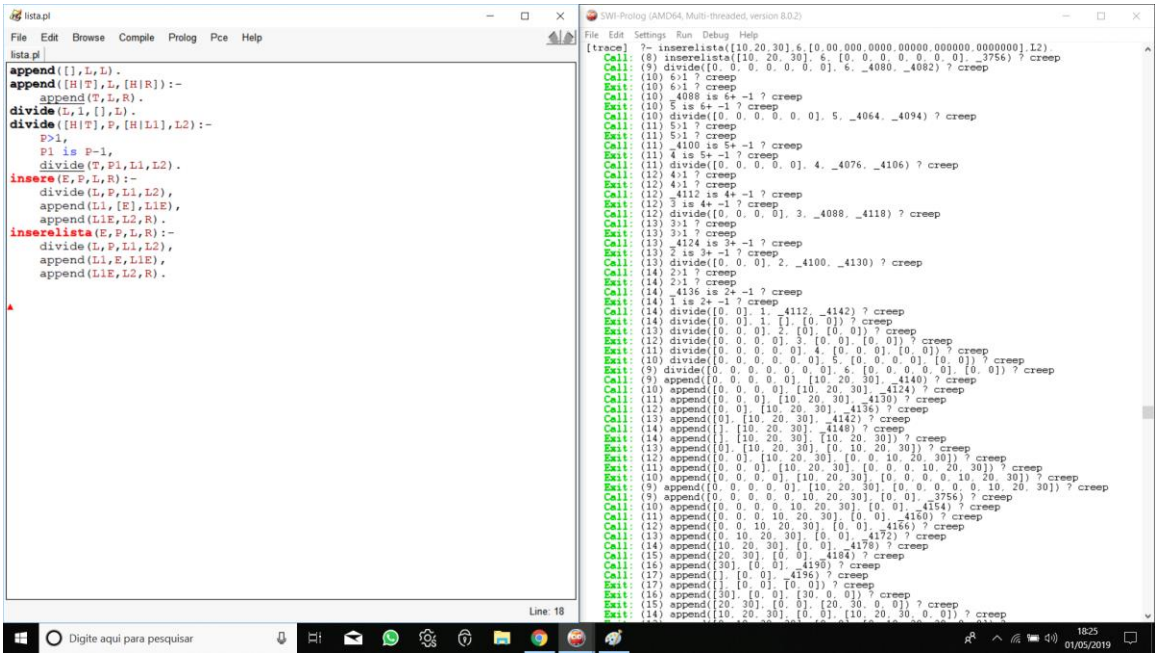
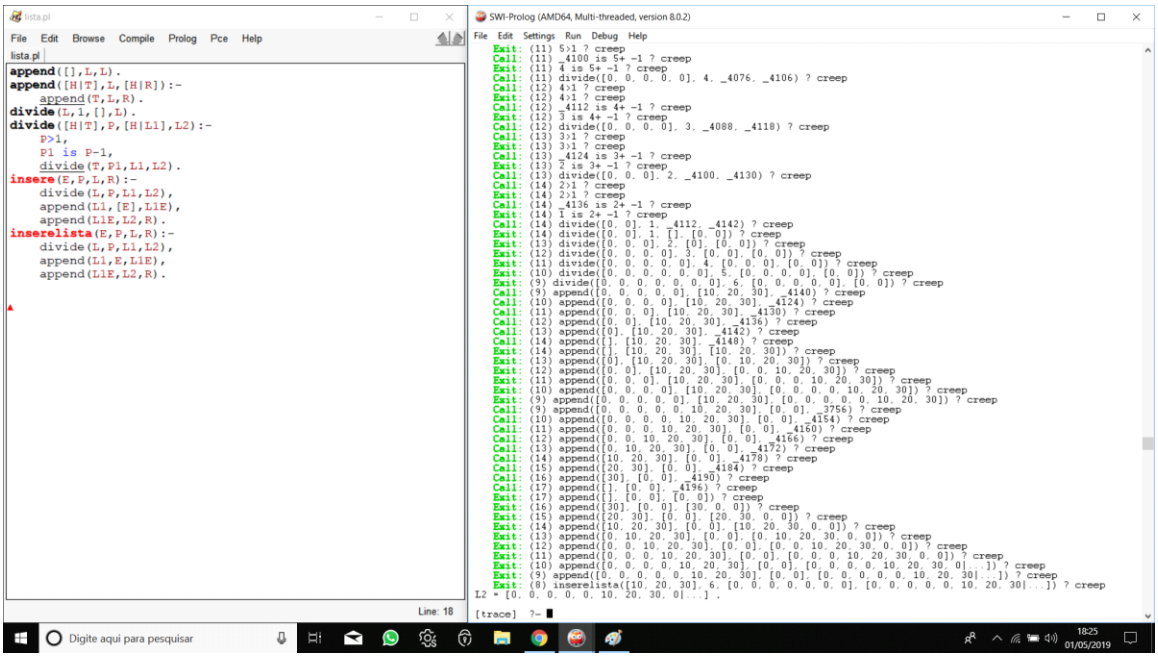
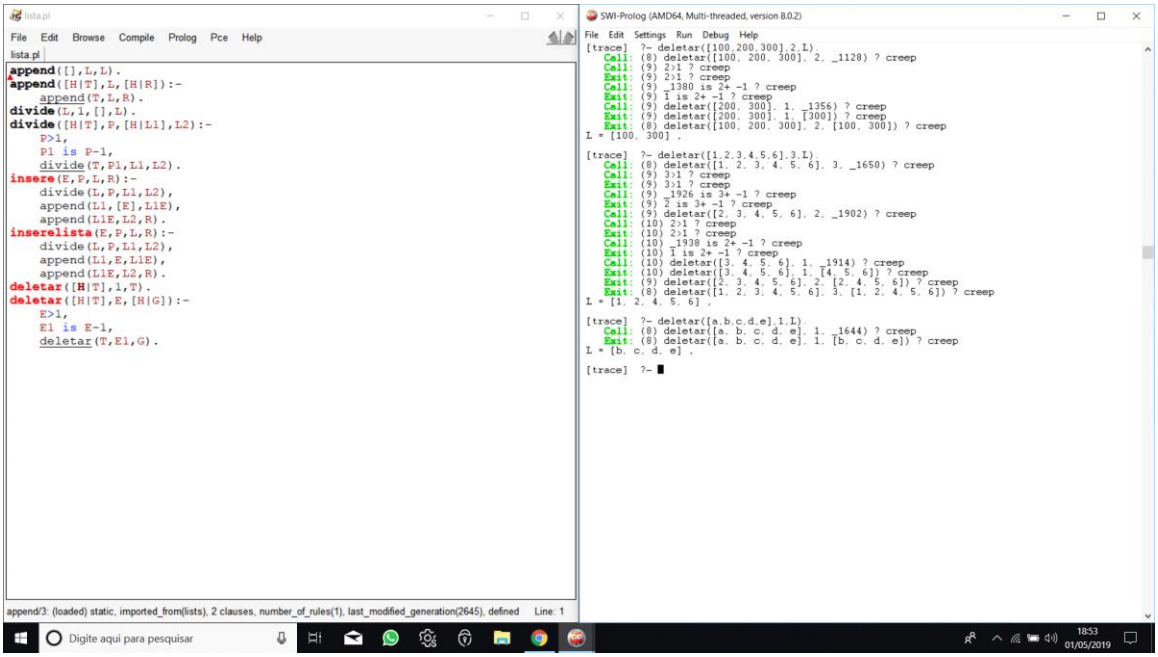

lista.pl [modified]
File Edit Browse Compile Prolog Pce Help
lista.pl [modified]
append([],L,L).
append([H|T],L,[H|R]):-
append(T,L,R).
divide(L,1,[],L).
divide([H|T],P,[H|L1],L2):-
P>1,
P1 is P-1,
divide(T,P1,L1,L2).
insere(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,[E],L1E),
append(L1E,L2,R).
Colourising buffer ... done, 0,00 seconds, 51 fragments
Line: 13
SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
% c:/users/arthur/documents/prolog/lista compiled 0.00 sec, 3 clauses
?- insere(8,1,[1,2,3],R).
R = [8, 1, 2, 3] .
?- insere(5,2,[a,b,c,d,e],R).
R = [a, 5, b, c, d, e] .
?- trace.
true.
[trace] ?- insere(8,1,[1,2,3],R).
Call: (8) insere(8, 1, [1, 2, 3], _3014) ? creep
Call: (9) divide([1, 2, 3], 1, _3258, _3260) ? creep
Exit: (9) divide([1, 2, 3], 1, [], [1, 2, 3]) ? creep
Call: (9) append([], [8], _3264) ? creep
Exit: (9) append([], [8], [8]) ? creep
Call: (9) append([8], [1, 2, 3], _3014) ? creep
Call: (10) append([], [1, 2, 3], _3248) ? creep
Exit: (10) append([], [1, 2, 3], [1, 2, 3]) ? creep
Exit: (9) append([8], [1, 2, 3], [8, 1, 2, 3]) ? creep
Exit: (8) insere(8, 1, [1, 2, 3], [8, 1, 2, 3]) ? creep
R = [8, 1, 2, 3] .
[trace] ?- insere(5,2,[10,20,25,5],R).
Call: (8) insere(5, 2, [10, 20, 25, 5], _3020) ? creep
Call: (9) divide([10, 20, 25, 5], 2, _3276, _3278) ? creep
Call: (10) 2>1 ? creep
Exit: (10) 2>1 ? creep
Call: (10) _3284 is 2+ -1 ? creep
Exit: (10) 1 is 2+ -1 ? creep
Call: (10) divide([20, 25, 5], 1, _3260, _3290) ? creep
Exit: (10) divide([20, 25, 5], 1, [], [20, 25, 5]) ? creep
Exit: (9) divide([10, 20, 25, 5], 2, [10], [20, 25, 5]) ? creep
Call: (9) append([10], [5], _3294) ? creep
Call: (10) append([], [5], _3298) ? creep
Exit: (10) append([], [5], [5]) ? creep
Exit: (9) append([10], [5], [10, 5]) ? creep
Call: (9) append([10, 5], [20, 25, 5], _3020) ? creep
Call: (10) append([5], [20, 25, 5], _3284) ? creep
Call: (11) append([], [20, 25, 5], _3290) ? creep
Exit: (11) append([], [20, 25, 5], [20, 25, 5]) ? creep
Exit: (10) append([5], [20, 25, 5], [5, 20, 25, 5]) ? creep
Exit: (9) append([10, 5], [20, 25, 5], [10, 5, 20, 25, 5]) ? creep
Exit: (8) insere(5, 2, [10, 20, 25, 5], [10, 5, 20, 25, 5]) ? creep
R = [10, 5, 20, 25, 5] .
[trace] ?- insere(1000,3,[1,2],R).
Call: (8) insere(1000, 3, [1, 2], _3008) ? creep
Call: (9) divide([1, 2], 3, _3252, _3254) ? creep
Call: (10) 3>1 ? creep
Exit: (10) 3>1 ? creep
Call: (10) _3260 is 3+ -1 ? creep
Exit: (10) 2 is 3+ -1 ? creep
Call: (10) divide([2], 2, _3236, _3266) ? creep
Call: (11) 2>1 ? creep
Exit: (11) 2>1 ? creep
Call: (11) _3272 is 2+ -1 ? creep
Exit: (11) 1 is 2+ -1 ? creep
Call: (11) divide([], 1, _3248, _3278) ? creep
Digite aqui para pesquisar
18:14
01/05/2019


lista.pl [modified]
File Edit Browse Compile Prolog Pce Help
lista.pl [modified]
append([],L,L).
append([H|T],L,[H|R]):-
append(T,L,R).
divide(L,1,[],L).
divide([H|T],P,[H|L1],L2):-
P>1,
P1 is P-1,
divide(T,P1,L1,L2).
insere(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,[E],L1E),
append(L1E,L2,R).
Colourising buffer ... done, 0,00 seconds, 51 fragments
Line: 13
SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
Exit: (9) append([8], [1, 2, 3], [8, 1, 2, 3]) ? creep
Exit: (8) insere(8, 1, [1, 2, 3], [8, 1, 2, 3]) ? creep
R = [8, 1, 2, 3] .
[trace] ?- insere(5,2,[10,20,25,5],R).
Call: (8) insere(5, 2, [10, 20, 25, 5], _3020) ? creep
Call: (9) divide([10, 20, 25, 5], 2, _3276, _3278) ? creep
Call: (10) 2>1 ? creep
Exit: (10) 2>1 ? creep
Call: (10) _3284 is 2+ -1 ? creep
Exit: (10) 1 is 2+ -1 ? creep
Call: (10) divide([20, 25, 5], 1, _3260, _3290) ? creep
Exit: (10) divide([20, 25, 5], 1, [], [20, 25, 5]) ? creep
Exit: (9) divide([10, 20, 25, 5], 2, [10], [20, 25, 5]) ? creep
Call: (9) append([10], [5], _3294) ? creep
Call: (10) append([], [5], _3298) ? creep
Exit: (10) append([], [5], [5]) ? creep
Exit: (9) append([10], [5], [10, 5]) ? creep
Call: (9) append([10, 5], [20, 25, 5], _3020) ? creep
Call: (10) append([5], [20, 25, 5], _3284) ? creep
Call: (11) append([], [20, 25, 5], _3290) ? creep
Exit: (11) append([], [20, 25, 5], [20, 25, 5]) ? creep
Exit: (10) append([5], [20, 25, 5], [5, 20, 25, 5]) ? creep
Exit: (9) append([10, 5], [20, 25, 5], [10, 5, 20, 25, 5]) ? creep
Exit: (8) insere(5, 2, [10, 20, 25, 5], [10, 5, 20, 25, 5]) ? creep
R = [10, 5, 20, 25, 5] .
[trace] ?- insere(1000,3,[1,2],R).
Call: (8) insere(1000, 3, [1, 2], _3008) ? creep
Call: (9) divide([1, 2], 3, _3252, _3254) ? creep
Call: (10) 3>1 ? creep
Exit: (10) 3>1 ? creep
Call: (10) _3260 is 3+ -1 ? creep
Exit: (10) 2 is 3+ -1 ? creep
Call: (10) divide([2], 2, _3236, _3266) ? creep
Call: (11) 2>1 ? creep
Exit: (11) 2>1 ? creep
Call: (11) _3272 is 2+ -1 ? creep
Exit: (11) 1 is 2+ -1 ? creep
Call: (11) divide([], 1, _3248, _3278) ? creep
Exit: (11) divide([], 1, [], []) ? creep
Exit: (10) divide([2], 2, [2], []) ? creep
Exit: (9) divide([1, 2], 3, [1, 2], []) ? creep
Call: (9) append([1, 2], [1000], _3282) ? creep
Call: (10) append([2], [1000], _3266) ? creep
Call: (11) append([], [1000], _3272) ? creep
Exit: (11) append([], [1000], [1000]) ? creep
Exit: (10) append([2], [1000], [2, 1000]) ? creep
Exit: (9) append([1, 2], [1000], [1, 2, 1000]) ? creep
Call: (9) append([1, 2, 1000], [], _3008) ? creep
Call: (10) append([2, 1000], [], _3278) ? creep
Call: (11) append([1000], [], _3284) ? creep
Call: (12) append([], [], _3290) ? creep
Exit: (12) append([], [], []) ? creep
Exit: (11) append([1000], [], [1000]) ? creep
Exit: (10) append([2, 1000], [], [2, 1000]) ? creep
Exit: (9) append([1, 2, 1000], [], [1, 2, 1000]) ? creep
Exit: (8) insere(1000, 3, [1, 2], [1, 2, 1000]) ? creep
R = [1, 2, 1000].
Digite aqui para pesquisar
18:13
01/05/2019


lista.pl
File Edit Browse Compile Prolog Pce Help
lista.pl
append([],L,L).
append([H|T],L,[H|R]):-
append(T,L,R).
divide(L,1,[],L).
divide([H|T],P,[H|L1],L2):-
P>1,
P1 is P-1,
divide(T,P1,L1,L2).
insere(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,[E],L1E),
append(L1E,L2,R).
inserelista(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,E,L1E),
append(L1E,L2,R).
Line: 18
SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
[trace] ?- inserelista([8,8,8],2,[1,2,3],L2).
Call: (8) inserelista([8, 8, 8], 2, [1, 2, 3], _3596) ? creep
Call: (9) divide([1, 2, 3], 2, _3864, _3866) ? creep
Call: (10) 2>1 ? creep
Exit: (10) 2>1 ? creep
Call: (10) _3872 is 2+ -1 ? creep
Exit: (10) 1 is 2+ -1 ? creep
Call: (10) divide([2, 3], 1, _3848, _3878) ? creep
Exit: (10) divide([2, 3], 1, [], [2, 3]) ? creep
Call: (9) divide([1, 2, 3], 2, [1], [2, 3]) ? creep
Call: (9) append([1], [8, 8, 8], _3876) ? creep
Call: (10) append([], [8, 8, 8], _3860) ? creep
Exit: (10) append([], [8, 8, 8], [8, 8, 8]) ? creep
Exit: (9) append([1], [8, 8, 8], [1, 8, 8, 8]) ? creep
Call: (9) append([1, 8, 8, 8], [2, 3], _3596) ? creep
Call: (10) append([8, 8, 8], [2, 3], _3866) ? creep
Call: (11) append([8, 8], [2, 3], _3872) ? creep
Call: (12) append([8], [2, 3], _3878) ? creep
Call: (13) append([], [2, 3], _3884) ? creep
Exit: (13) append([], [2, 3], [2, 3]) ? creep
Exit: (12) append([8], [2, 3], [8, 2, 3]) ? creep
Exit: (11) append([8, 8], [2, 3], [8, 8, 2, 3]) ? creep
Exit: (10) append([8, 8, 8], [2, 3], [8, 8, 8, 2, 3]) ? creep
Exit: (9) append([1, 8, 8, 8], [2, 3], [1, 8, 8, 8, 2, 3]) ? creep
Exit: (8) inserelista([8, 8, 8], 2, [1, 2, 3], [1, 8, 8, 8, 2, 3]) ? creep
L2 = [1, 8, 8, 8, 2, 3] .
[trace] ?- inserelista([a,b,c,d,e],2,[1,2,3],L2).
Call: (8) inserelista([a, b, c, d, e], 2, [1, 2, 3], _3608) ? creep
Call: (9) divide([1, 2, 3], 2, _3906, _3908) ? creep
Call: (10) 2>1 ? creep
Exit: (10) 2>1 ? creep
Call: (10) _3914 is 2+ -1 ? creep
Exit: (10) 1 is 2+ -1 ? creep
Call: (10) divide([2, 3], 1, _3890, _3920) ? creep
Exit: (10) divide([2, 3], 1, [], [2, 3]) ? creep
Exit: (9) divide([1, 2, 3], 2, [1], [2, 3]) ? creep
Call: (9) append([1], [a, b, c, d, e], _3918) ? creep
Call: (10) append([], [a, b, c, d, e], _3902) ? creep
Exit: (10) append([], [a, b, c, d, e], [a, b, c, d, e]) ? creep
Exit: (9) append([1], [a, b, c, d, e], [1, a, b, c, d, e]) ? creep
Call: (9) append([1, a, b, c, d, e], [2, 3], _3608) ? creep
Call: (10) append([a, b, c, d, e], [2, 3], _3908) ? creep
Call: (11) append([b, c, d, e], [2, 3], _3914) ? creep
Call: (12) append([c, d, e], [2, 3], _3920) ? creep
Call: (13) append([d, e], [2, 3], _3926) ? creep
Call: (14) append([e], [2, 3], _3932) ? creep
Call: (15) append([], [2, 3], _3938) ? creep
Exit: (15) append([], [2, 3], [2, 3]) ? creep
Exit: (14) append([e], [2, 3], [e, 2, 3]) ? creep
Exit: (13) append([d, e], [2, 3], [d, e, 2, 3]) ? creep
Exit: (12) append([c, d, e], [2, 3], [c, d, e, 2, 3]) ? creep
Exit: (11) append([b, c, d, e], [2, 3], [b, c, d, e, 2, 3]) ? creep
Exit: (10) append([a, b, c, d, e], [2, 3], [a, b, c, d, e, 2, 3]) ? creep
Exit: (9) append([1, a, b, c, d, e], [2, 3], [1, a, b, c, d, e, 2, 3]) ? creep
Exit: (8) inserelista([a, b, c, d, e], 2, [1, 2, 3], [1, a, b, c, d, e, 2, 3]) ? creep
L2 = [1, a, b, c, d, e, 2, 3] .
Digite aqui para pesquisar
18:25
01/05/2019


lista.pl
File Edit Browse Compile Prolog Pce Help
lista.pl
append([],L,L).
append([H|T],L,[H|R]):-
append(T,L,R).
divide(L,1,[],L).
divide([H|T],P,[H|L1],L2):-
P>1,
P1 is P-1,
divide(T,P1,L1,L2).
insere(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,[E],L1E),
append(L1E,L2,R).
inserelista(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,E,L1E),
append(L1E,L2,R).
Line: 18
SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
[trace] ?- inserelista([10,20,30],6,[0,00,000,0000,00000,000000,0000000],L2).
Call: (8) inserelista([10, 20, 30], 6, [0, 0, 0, 0, 0, 0, 0], _3756) ? creep
Call: (9) divide([0, 0, 0, 0, 0, 0, 0], 6, _4080, _4082) ? creep
Call: (10) 6>1 ? creep
Exit: (10) 6>1 ? creep
Call: (10) _4088 is 6+ -1 ? creep
Exit: (10) 5 is 6+ -1 ? creep
Call: (10) divide([0, 0, 0, 0, 0, 0], 5, _4064, _4094) ? creep
Call: (11) 5>1 ? creep
Exit: (11) 5>1 ? creep
Call: (11) _4100 is 5+ -1 ? creep
Exit: (11) 4 is 5+ -1 ? creep
Call: (11) divide([0, 0, 0, 0, 0], 4, _4076, _4106) ? creep
Call: (12) 4>1 ? creep
Exit: (12) 4>1 ? creep
Call: (12) _4112 is 4+ -1 ? creep
Exit: (12) 3 is 4+ -1 ? creep
Call: (12) divide([0, 0, 0, 0], 3, _4088, _4118) ? creep
Call: (13) 3>1 ? creep
Exit: (13) 3>1 ? creep
Call: (13) _4124 is 3+ -1 ? creep
Exit: (13) 2 is 3+ -1 ? creep
Call: (13) divide([0, 0, 0], 2, _4100, _4130) ? creep
Call: (14) 2>1 ? creep
Exit: (14) 2>1 ? creep
Call: (14) _4136 is 2+ -1 ? creep
Exit: (14) 1 is 2+ -1 ? creep
Call: (14) divide([0, 0], 1, _4112, _4142) ? creep
Exit: (14) divide([0, 0], 1, [], [0, 0]) ? creep
Exit: (13) divide([0, 0, 0], 2, [0], [0, 0]) ? creep
Exit: (12) divide([0, 0, 0, 0], 3, [0, 0], [0, 0]) ? creep
Exit: (11) divide([0, 0, 0, 0, 0], 4, [0, 0, 0], [0, 0]) ? creep
Exit: (10) divide([0, 0, 0, 0, 0, 0], 5, [0, 0, 0, 0], [0, 0]) ? creep
Exit: (9) divide([0, 0, 0, 0, 0, 0, 0], 6, [0, 0, 0, 0, 0], [0, 0]) ? creep
Call: (9) append([0, 0, 0, 0, 0], [10, 20, 30], _4140) ? creep
Call: (10) append([0, 0, 0, 0], [10, 20, 30], _4124) ? creep
Call: (11) append([0, 0, 0], [10, 20, 30], _4130) ? creep
Call: (12) append([0, 0], [10, 20, 30], _4136) ? creep
Call: (13) append([0], [10, 20, 30], _4142) ? creep
Call: (14) append([], [10, 20, 30], _4148) ? creep
Exit: (14) append([], [10, 20, 30], [10, 20, 30]) ? creep
Exit: (13) append([0], [10, 20, 30], [0, 10, 20, 30]) ? creep
Exit: (12) append([0, 0], [10, 20, 30], [0, 0, 10, 20, 30]) ? creep
Exit: (11) append([0, 0, 0], [10, 20, 30], [0, 0, 0, 10, 20, 30]) ? creep
Exit: (10) append([0, 0, 0, 0], [10, 20, 30], [0, 0, 0, 0, 10, 20, 30]) ? creep
Exit: (9) append([0, 0, 0, 0, 0], [10, 20, 30], [0, 0, 0, 0, 0, 10, 20, 30]) ? creep
Call: (9) append([0, 0, 0, 0, 10, 20, 30], [0, 0], _3756) ? creep
Call: (10) append([0, 0, 0, 10, 20, 30], [0, 0], _4154) ? creep
Call: (11) append([0, 0, 10, 20, 30], [0, 0], _4160) ? creep
Call: (12) append([0, 10, 20, 30], [0, 0], _4166) ? creep
Call: (13) append([10, 20, 30], [0, 0], _4172) ? creep
Call: (14) append([20, 30], [0, 0], _4178) ? creep
Call: (15) append([30], [0, 0], _4184) ? creep
Call: (16) append([], [0, 0], _4190) ? creep
Exit: (16) append([], [0, 0], [0, 0]) ? creep
Exit: (15) append([30], [0, 0], [30, 0, 0]) ? creep
Exit: (14) append([20, 30], [0, 0], [20, 30, 0, 0]) ? creep
Exit: (13) append([10, 20, 30], [0, 0], [10, 20, 30, 0, 0]) ? creep
Digite aqui para pesquisar
18:25
01/05/2019


lista.pl
File Edit Browse Compile Prolog Pce Help
lista.pl
append([],L,L).
append([H|T],L,[H|R]):-
append(T,L,R).
divide(L,1,[],L).
divide([H|T],P,[H|L1],L2):-
P>1,
P1 is P-1,
divide(T,P1,L1,L2).
insere(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,[E],L1E),
append(L1E,L2,R).
inserelista(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,E,L1E),
append(L1E,L2,R).
Line: 18
SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
Exit: (11) 5>1 ? creep
Call: (11) _4100 is 5+ -1 ? creep
Exit: (11) 4 is 5+ -1 ? creep
Call: (11) divide([0, 0, 0, 0, 0], 4, _4076, _4106) ? creep
Call: (12) 4>1 ? creep
Exit: (12) 4>1 ? creep
Call: (12) _4112 is 4+ -1 ? creep
Exit: (12) 3 is 4+ -1 ? creep
Call: (12) divide([0, 0, 0, 0], 3, _4088, _4118) ? creep
Call: (13) 3>1 ? creep
Exit: (13) 3>1 ? creep
Call: (13) _4124 is 3+ -1 ? creep
Exit: (13) 2 is 3+ -1 ? creep
Call: (13) divide([0, 0, 0], 2, _4100, _4130) ? creep
Call: (14) 2>1 ? creep
Exit: (14) 2>1 ? creep
Call: (14) _4136 is 2+ -1 ? creep
Exit: (14) 1 is 2+ -1 ? creep
Call: (14) divide([0, 0], 1, _4112, _4142) ? creep
Exit: (14) divide([0, 0], 1, [], [0, 0]) ? creep
Exit: (13) divide([0, 0, 0], 2, [0], [0, 0]) ? creep
Exit: (12) divide([0, 0, 0, 0], 3, [0, 0], [0, 0]) ? creep
Exit: (11) divide([0, 0, 0, 0, 0], 4, [0, 0, 0], [0, 0]) ? creep
Exit: (10) divide([0, 0, 0, 0, 0, 0], 5, [0, 0, 0, 0], [0, 0]) ? creep
Exit: (9) divide([0, 0, 0, 0, 0, 0, 0], 6, [0, 0, 0, 0, 0], [0, 0]) ? creep
Call: (9) append([0, 0, 0, 0, 0], [10, 20, 30], _4140) ? creep
Call: (10) append([0, 0, 0, 0], [10, 20, 30], _4124) ? creep
Call: (11) append([0, 0, 0], [10, 20, 30], _4130) ? creep
Call: (12) append([0, 0], [10, 20, 30], _4136) ? creep
Call: (13) append([0], [10, 20, 30], _4142) ? creep
Call: (14) append([], [10, 20, 30], _4148) ? creep
Exit: (14) append([], [10, 20, 30], [10, 20, 30]) ? creep
Exit: (13) append([0], [10, 20, 30], [0, 10, 20, 30]) ? creep
Exit: (12) append([0, 0], [10, 20, 30], [0, 0, 10, 20, 30]) ? creep
Exit: (11) append([0, 0, 0], [10, 20, 30], [0, 0, 0, 10, 20, 30]) ? creep
Exit: (10) append([0, 0, 0, 0], [10, 20, 30], [0, 0, 0, 0, 10, 20, 30]) ? creep
Exit: (9) append([0, 0, 0, 0, 0], [10, 20, 30], [0, 0, 0, 0, 0, 10, 20, 30]) ? creep
Call: (9) append([0, 0, 0, 0, 0, 10, 20, 30], [0, 0], _3756) ? creep
Call: (10) append([0, 0, 0, 0, 10, 20, 30], [0, 0], _4154) ? creep
Call: (11) append([0, 0, 0, 10, 20, 30], [0, 0], _4160) ? creep
Call: (12) append([0, 0, 10, 20, 30], [0, 0], _4166) ? creep
Call: (13) append([0, 10, 20, 30], [0, 0], _4172) ? creep
Call: (14) append([10, 20, 30], [0, 0], _4178) ? creep
Call: (15) append([20, 30], [0, 0], _4184) ? creep
Call: (16) append([30], [0, 0], _4190) ? creep
Call: (17) append([], [0, 0], _4196) ? creep
Exit: (17) append([], [0, 0], [0, 0]) ? creep
Exit: (16) append([30], [0, 0], [30, 0, 0]) ? creep
Exit: (15) append([20, 30], [0, 0], [20, 30, 0, 0]) ? creep
Exit: (14) append([10, 20, 30], [0, 0], [10, 20, 30, 0, 0]) ? creep
Exit: (13) append([0, 10, 20, 30], [0, 0], [0, 10, 20, 30, 0, 0]) ? creep
Exit: (12) append([0, 0, 10, 20, 30], [0, 0], [0, 0, 10, 20, 30, 0, 0]) ? creep
Exit: (11) append([0, 0, 0, 10, 20, 30], [0, 0], [0, 0, 0, 10, 20, 30, 0, 0]) ? creep
Exit: (10) append([0, 0, 0, 0, 10, 20, 30], [0, 0], [0, 0, 0, 0, 10, 20, 30, 0|...]) ? creep
Exit: (9) append([0, 0, 0, 0, 0, 10, 20, 30], [0, 0], [0, 0, 0, 0, 0, 10, 20, 30|...]) ? creep
Exit: (8) inserelista([10, 20, 30], 6, [0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 10, 20, 30|...]) ? creep
L2 = [0, 0, 0, 0, 10, 20, 30, 0|...].
[trace] ?-
Digite aqui para pesquisar
18:25
01/05/2019


lista.pl
File Edit Browse Compile Prolog Pce Help
lista.pl
append([],L,L).
append([H|T],L,[H|R]):-
append(T,L,R).
divide(L,1,[],L).
divide([H|T],P,[H|L1],L2):-
P>1,
P1 is P-1,
divide(T,P1,L1,L2).
insere(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,[E],L1E),
append(L1E,L2,R).
inserelista(E,P,L,R):-
divide(L,P,L1,L2),
append(L1,E,L1E),
append(L1E,L2,R).
deletar([H|T],1,T).
deletar([H|T],E,[H|G]):-
E>1,
E1 is E-1,
deletar(T,E1,G).
append/3: (loaded) static, imported_from(lists), 2 clauses, number_of_rules(1), last_modified_generation(2645), defined
Line: 1
SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
[trace] ?- deletar([100,200,300],2,L).
Call: (8) deletar([100, 200, 300], 2, _1128) ? creep
Call: (9) 2>1 ? creep
Exit: (9) 2>1 ? creep
Call: (9) _1380 is 2+ -1 ? creep
Exit: (9) 1 is 2+ -1 ? creep
Call: (9) deletar([200, 300], 1, _1356) ? creep
Exit: (9) deletar([200, 300], 1, [300]) ? creep
Exit: (8) deletar([100, 200, 300], 2, [100, 300]) ? creep
L = [100, 300] .
[trace] ?- deletar([1,2,3,4,5,6],3,L).
Call: (8) deletar([1, 2, 3, 4, 5, 6], 3, _1650) ? creep
Call: (9) 3>1 ? creep
Exit: (9) 3>1 ? creep
Call: (9) _1926 is 3+ -1 ? creep
Exit: (9) 2 is 3+ -1 ? creep
Call: (9) deletar([2, 3, 4, 5, 6], 2, _1902) ? creep
Call: (10) 2>1 ? creep
Exit: (10) 2>1 ? creep
Call: (10) _1938 is 2+ -1 ? creep
Exit: (10) 1 is 2+ -1 ? creep
Call: (10) deletar([3, 4, 5, 6], 1, _1914) ? creep
Exit: (10) deletar([3, 4, 5, 6], 1, [4, 5, 6]) ? creep
Exit: (9) deletar([2, 3, 4, 5, 6], 2, [2, 4, 5, 6]) ? creep
Exit: (8) deletar([1, 2, 3, 4, 5, 6], 3, [1, 2, 4, 5, 6]) ? creep
L = [1, 2, 4, 5, 6] .
[trace] ?- deletar([a,b,c,d,e],1,L).
Call: (8) deletar([a, b, c, d, e], 1, _1644) ? creep
Exit: (8) deletar([a, b, c, d, e], 1, [b, c, d, e]) ? creep
L = [b, c, d, e] .
[trace] ?-
Digite aqui para pesquisar
18:53
01/05/2019

%append(+Lista1, +Lista2, -ListaConcatenada).

%appeend([1,2],[3,4],L).

%L = [1,2,3,4].

%insere(+Elemento, +Posiçao, +Lista, -Listainserida).

%insere(4,2,[1,2,3,4,5],L).

%L = [1,4,2,3,4,5].

%inserelista(+Elemento ou Lista, +Posiçao, +Lista, -ListaInserida).

%inserelista([1,2,3],3,[1,2,3,4,5],L).

%L = [1,2,1,2,3,3,4,5].

%deletar(+Lista, +Posiçao, -ListaRemovida).

%deletar([1,20,13,4,52],2,L).

%L = [1,13,4,52].