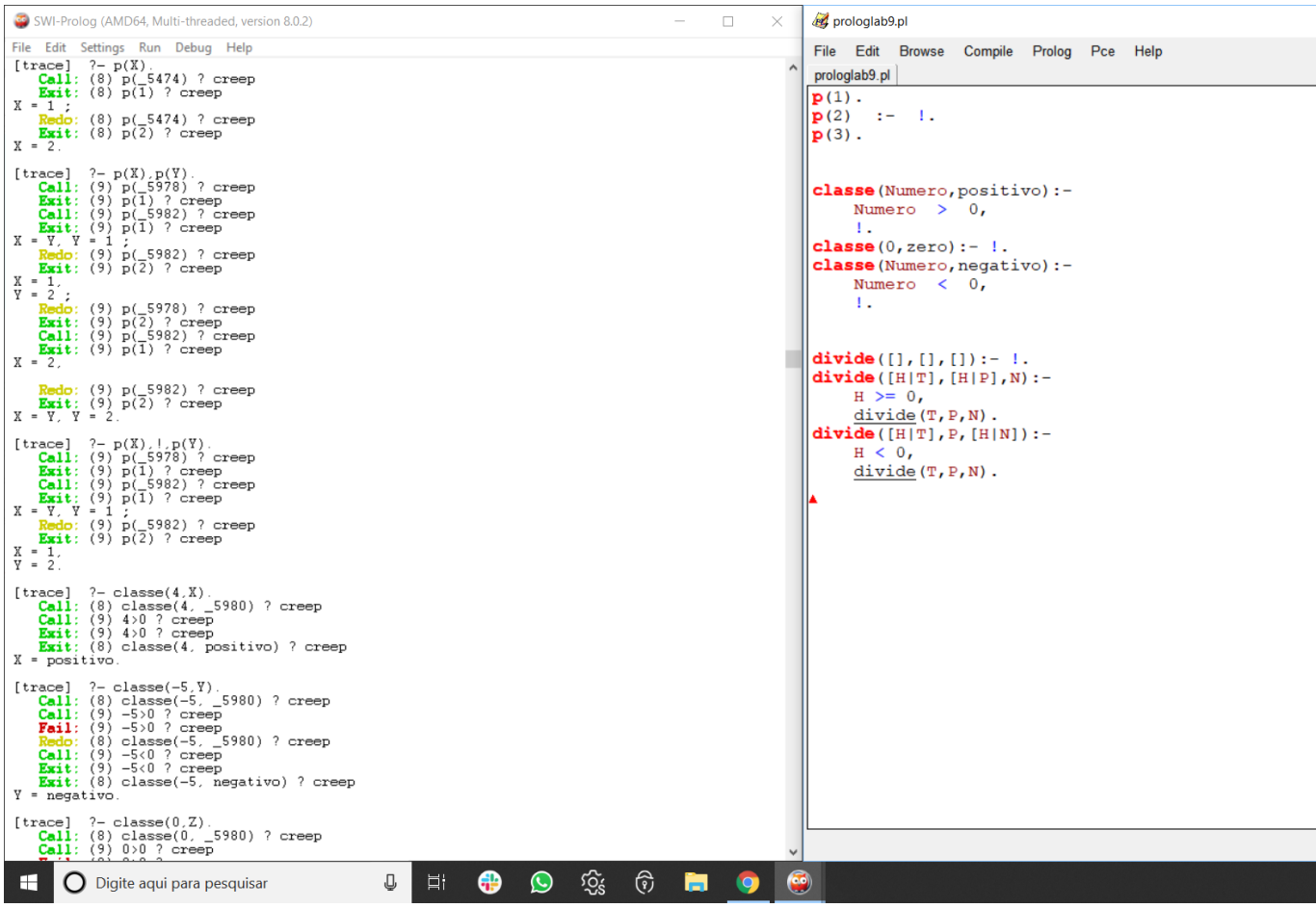
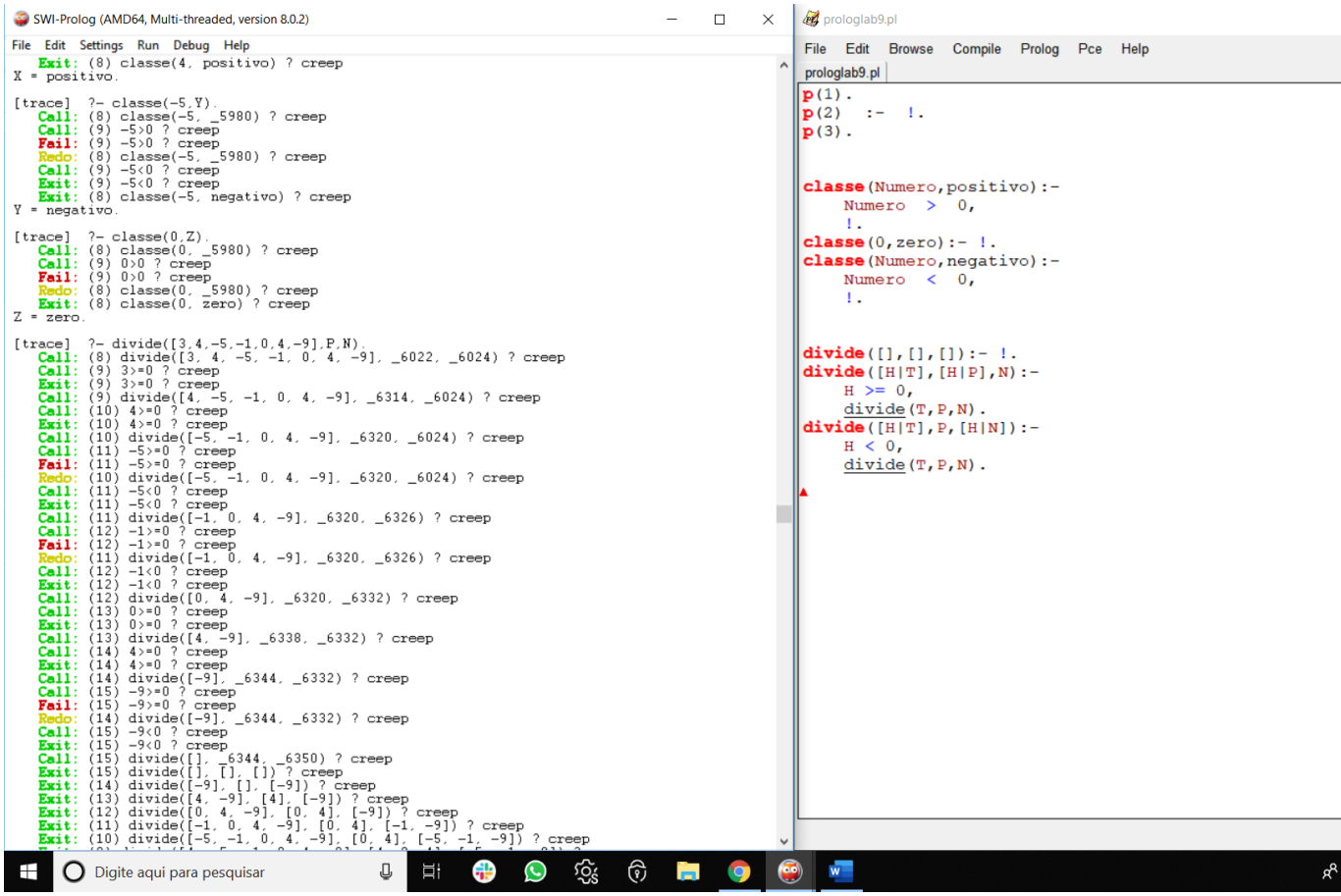
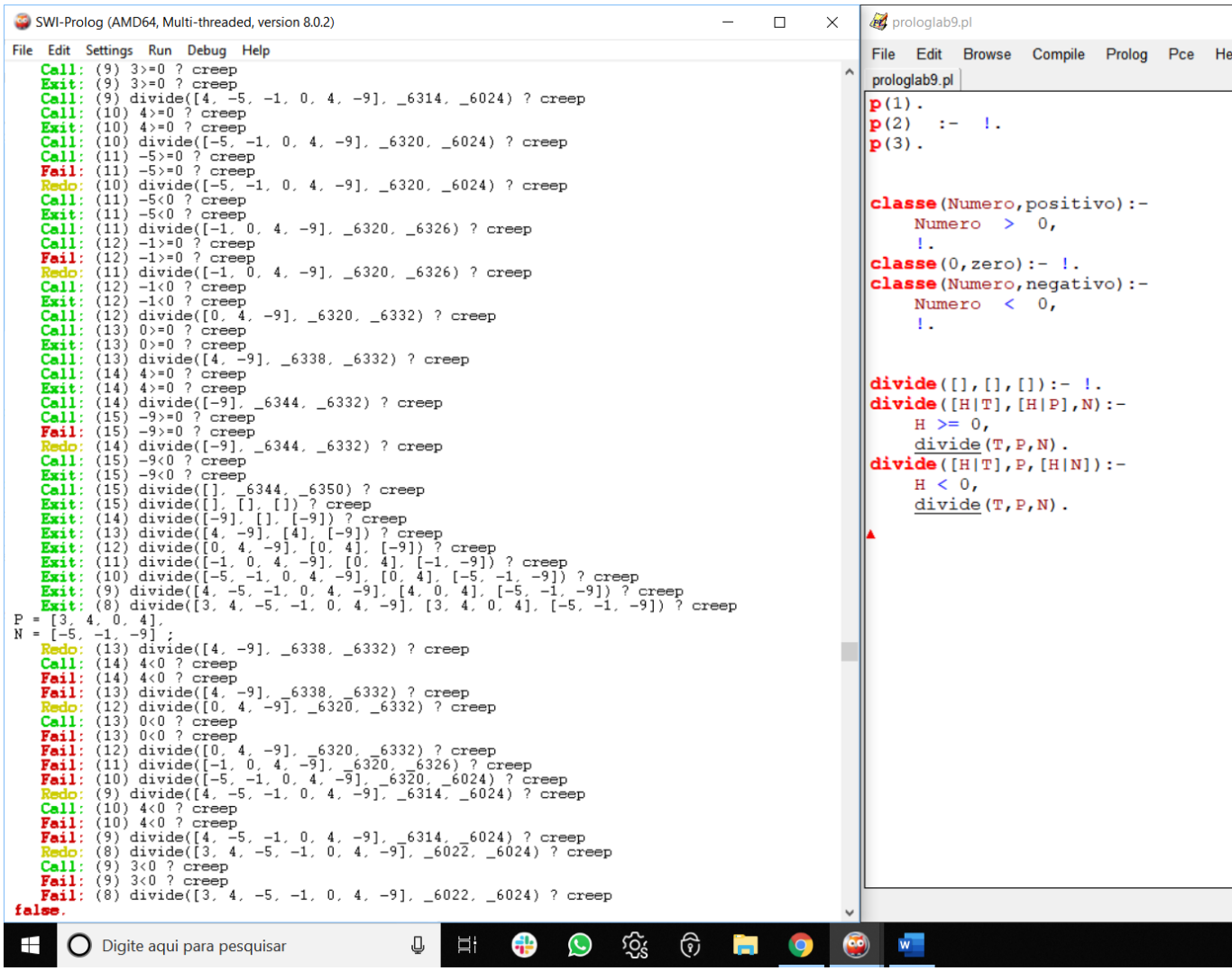

SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
[trace] ?- p(X).
Call: (8) p(_5474) ? creep
Exit: (8) p(1) ? creep
X = 1 ;
Redo: (8) p(_5474) ? creep
Exit: (8) p(2) ? creep
X = 2.
[trace] ?- p(X),p(Y).
Call: (9) p(_5978) ? creep
Exit: (9) p(1) ? creep
Call: (9) p(_5982) ? creep
Exit: (9) p(1) ? creep
X = Y, Y = 1 ;
Redo: (9) p(_5982) ? creep
Exit: (9) p(2) ? creep
X = 1,
Y = 2 ;
Redo: (9) p(_5978) ? creep
Exit: (9) p(2) ? creep
Call: (9) p(_5982) ? creep
Exit: (9) p(1) ? creep
X = 2,
Redo: (9) p(_5982) ? creep
Exit: (9) p(2) ? creep
X = Y, Y = 2.
[trace] ?- p(X),!,p(Y).
Call: (9) p(_5978) ? creep
Exit: (9) p(1) ? creep
Call: (9) p(_5982) ? creep
Exit: (9) p(1) ? creep
X = Y, Y = 1 ;
Redo: (9) p(_5982) ? creep
Exit: (9) p(2) ? creep
X = 1,
Y = 2.
[trace] ?- classe(4,X).
Call: (8) classe(4, _5980) ? creep
Call: (9) 4>0 ? creep
Exit: (9) 4>0 ? creep
Exit: (8) classe(4, positivo) ? creep
X = positivo.
[trace] ?- classe(-5,Y).
Call: (8) classe(-5, _5980) ? creep
Call: (9) -5>0 ? creep
Fail: (9) -5>0 ? creep
Redo: (8) classe(-5, _5980) ? creep
Call: (9) -5<0 ? creep
Exit: (9) -5<0 ? creep
Exit: (8) classe(-5, negativo) ? creep
Y = negativo.
[trace] ?- classe(0,Z).
Call: (8) classe(0, _5980) ? creep
Call: (9) 0>0 ? creep
prologlab9.pl
File Edit Browse Compile Prolog Pce Help
prologlab9.pl
p(1).
p(2) :- !.
p(3).
classe(Numero,positivo):-
Numero > 0,
!.
classe(0,zero):- !.
classe(Numero,negativo):-
Numero < 0,
!.
divide([],[],[]):- !.
divide([H|T],[H|P],N):-
H >= 0,
divide(T,P,N).
divide([H|T],P,[H|N]):-
H < 0,
divide(T,P,N).
Digite aqui para pesquisar


SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
Exit: (8) classe(4, positivo) ? creep
X = positivo.
[trace] ?- classe(-5,Y).
Call: (8) classe(-5, _5980) ? creep
Call: (9) -5>0 ? creep
Fail: (9) -5>0 ? creep
Redo: (8) classe(-5, _5980) ? creep
Call: (9) -5<0 ? creep
Exit: (9) -5<0 ? creep
Exit: (8) classe(-5, negativo) ? creep
Y = negativo.
[trace] ?- classe(0,Z).
Call: (8) classe(0, _5980) ? creep
Call: (9) 0>0 ? creep
Fail: (9) 0>0 ? creep
Redo: (8) classe(0, _5980) ? creep
Exit: (8) classe(0, zero) ? creep
Z = zero.
[trace] ?- divide([3,4,-5,-1,0,4,-9],P,N).
Call: (8) divide([3, 4, -5, -1, 0, 4, -9], _6022, _6024) ? creep
Call: (9) 3>=0 ? creep
Exit: (9) 3>=0 ? creep
Call: (9) divide([4, -5, -1, 0, 4, -9], _6314, _6024) ? creep
Call: (10) 4>=0 ? creep
Exit: (10) 4>=0 ? creep
Call: (10) divide([-5, -1, 0, 4, -9], _6320, _6024) ? creep
Call: (11) -5>=0 ? creep
Fail: (11) -5>=0 ? creep
Redo: (10) divide([-5, -1, 0, 4, -9], _6320, _6024) ? creep
Call: (11) -5<0 ? creep
Exit: (11) -5<0 ? creep
Call: (11) divide([-1, 0, 4, -9], _6320, _6326) ? creep
Call: (12) -1>=0 ? creep
Fail: (12) -1>=0 ? creep
Redo: (11) divide([-1, 0, 4, -9], _6320, _6326) ? creep
Call: (12) -1<0 ? creep
Exit: (12) -1<0 ? creep
Call: (12) divide([0, 4, -9], _6320, _6332) ? creep
Call: (13) 0>=0 ? creep
Exit: (13) 0>=0 ? creep
Call: (13) divide([4, -9], _6338, _6332) ? creep
Call: (14) 4>=0 ? creep
Exit: (14) 4>=0 ? creep
Call: (14) divide([-9], _6344, _6332) ? creep
Call: (15) -9>=0 ? creep
Fail: (15) -9>=0 ? creep
Redo: (14) divide([-9], _6344, _6332) ? creep
Call: (15) -9<0 ? creep
Exit: (15) -9<0 ? creep
Call: (15) divide([], _6344, _6350) ? creep
Exit: (15) divide([], [], []) ? creep
Exit: (14) divide([-9], [], [-9]) ? creep
Exit: (13) divide([4, -9], [4], [-9]) ? creep
Exit: (12) divide([0, 4, -9], [0, 4], [-9]) ? creep
Exit: (11) divide([-1, 0, 4, -9], [0, 4], [-1, -9]) ? creep
Exit: (10) divide([-5, -1, 0, 4, -9], [0, 4], [-5, -1, -9]) ? creep
prologlab9.pl
File Edit Browse Compile Prolog Pce Help
prologlab9.pl
p(1).
p(2) :- !.
p(3).
classe(Numero,positivo):-
Numero > 0,
!.
classe(0,zero):- !.
classe(Numero,negativo):-
Numero < 0,
!.
divide([],[],[]):- !.
divide([H|T],[H|P],N):-
H >= 0,
divide(T,P,N).
divide([H|T],P,[H|N]):-
H < 0,
divide(T,P,N).
Digite aqui para pesquisar



2- O programa indica se o número digitado é positivo, zero ou negativo.

Quando já testado em positivo, não é necessário testar em zero ou em negativo (pois foi confirmado que o número é maior que zero).

O mesmo para zero e negativo.