**Solve Sudoku using Prolog**

**Code:**

% to run the code in SWI-Prolog, do

% problem(1, Rows), sudoku(Rows), maplist(portray\_clause, Rows).

:- use\_module(library(clpfd)).

sudoku(Rows) :-

length(Rows, 9), maplist(same\_length(Rows), Rows),

append(Rows, Vs), Vs ins 1..9,

maplist(all\_distinct, Rows),

transpose(Rows, Columns),

maplist(all\_distinct, Columns),

Rows = [As,Bs,Cs,Ds,Es,Fs,Gs,Hs,Is],

blocks(As, Bs, Cs),

blocks(Ds, Es, Fs),

blocks(Gs, Hs, Is).

blocks([], [], []).

blocks([N1,N2,N3|Ns1], [N4,N5,N6|Ns2], [N7,N8,N9|Ns3]) :-

all\_distinct([N1,N2,N3,N4,N5,N6,N7,N8,N9]),

blocks(Ns1, Ns2, Ns3).

problem(0, [[\_,\_,\_,\_,\_,\_,\_,\_,\_],

[\_,\_,\_,\_,\_,3,\_,8,5],

[\_,\_,1,\_,2,\_,\_,\_,\_],

[\_,\_,\_,5,\_,7,\_,\_,\_],

[\_,\_,4,\_,\_,\_,1,\_,\_],

[\_,9,\_,\_,\_,\_,\_,\_,\_],

[5,\_,\_,\_,\_,\_,\_,7,3],

[\_,\_,2,\_,1,\_,\_,\_,\_],

[\_,\_,\_,\_,4,\_,\_,\_,9]]).

problem(1, P) :-

P = [[1,\_,\_,8,\_,4,\_,\_,\_],

[\_,2,\_,\_,\_,\_,4,5,6],

[\_,\_,3,2,\_,5,\_,\_,\_],

[\_,\_,\_,4,\_,\_,8,\_,5],

[7,8,9,\_,5,\_,\_,\_,\_],

[\_,\_,\_,\_,\_,6,2,\_,3],

[8,\_,1,\_,\_,\_,7,\_,\_],

[\_,\_,\_,1,2,3,\_,8,\_],

[2,\_,5,\_,\_,\_,\_,\_,9]].

problem(2, P) :-

P = [[\_,\_,2,\_,3,\_,1,\_,\_],

[\_,4,\_,\_,\_,\_,\_,3,\_],

[1,\_,5,\_,\_,\_,\_,8,2],

[\_,\_,\_,2,\_,\_,6,5,\_],

[9,\_,\_,\_,8,7,\_,\_,3],

[\_,\_,\_,\_,4,\_,\_,\_,\_],

[8,\_,\_,\_,7,\_,\_,\_,4],

[\_,9,3,1,\_,\_,\_,6,\_],

[\_,\_,7,\_,6,\_,5,\_,\_]].

problem(3, P) :-

P = [[1,\_,\_,\_,\_,\_,\_,\_,\_],

[\_,\_,2,7,4,\_,\_,\_,\_],

[\_,\_,\_,5,\_,\_,\_,\_,4],

[\_,3,\_,\_,\_,\_,\_,\_,\_],

[7,5,\_,\_,\_,\_,\_,\_,\_],

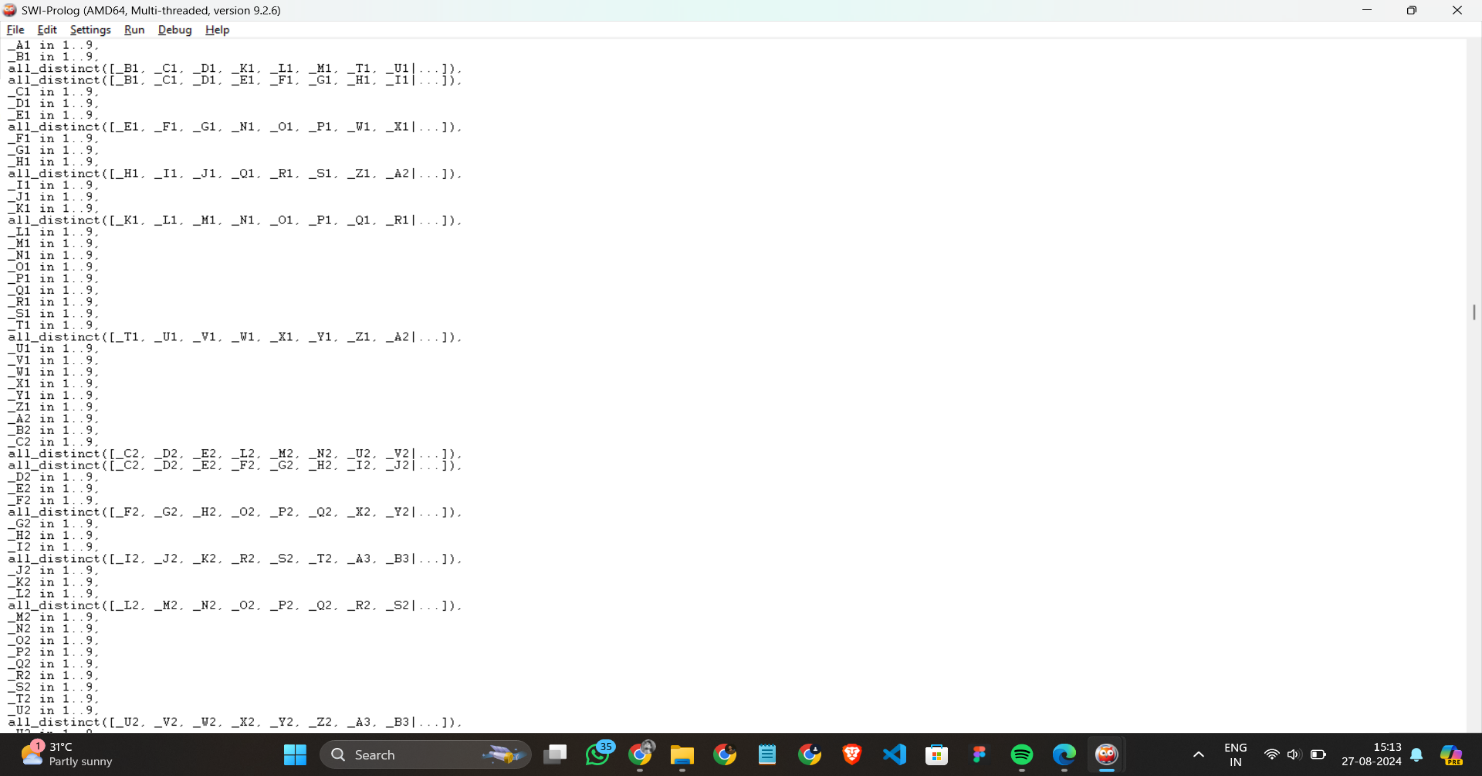
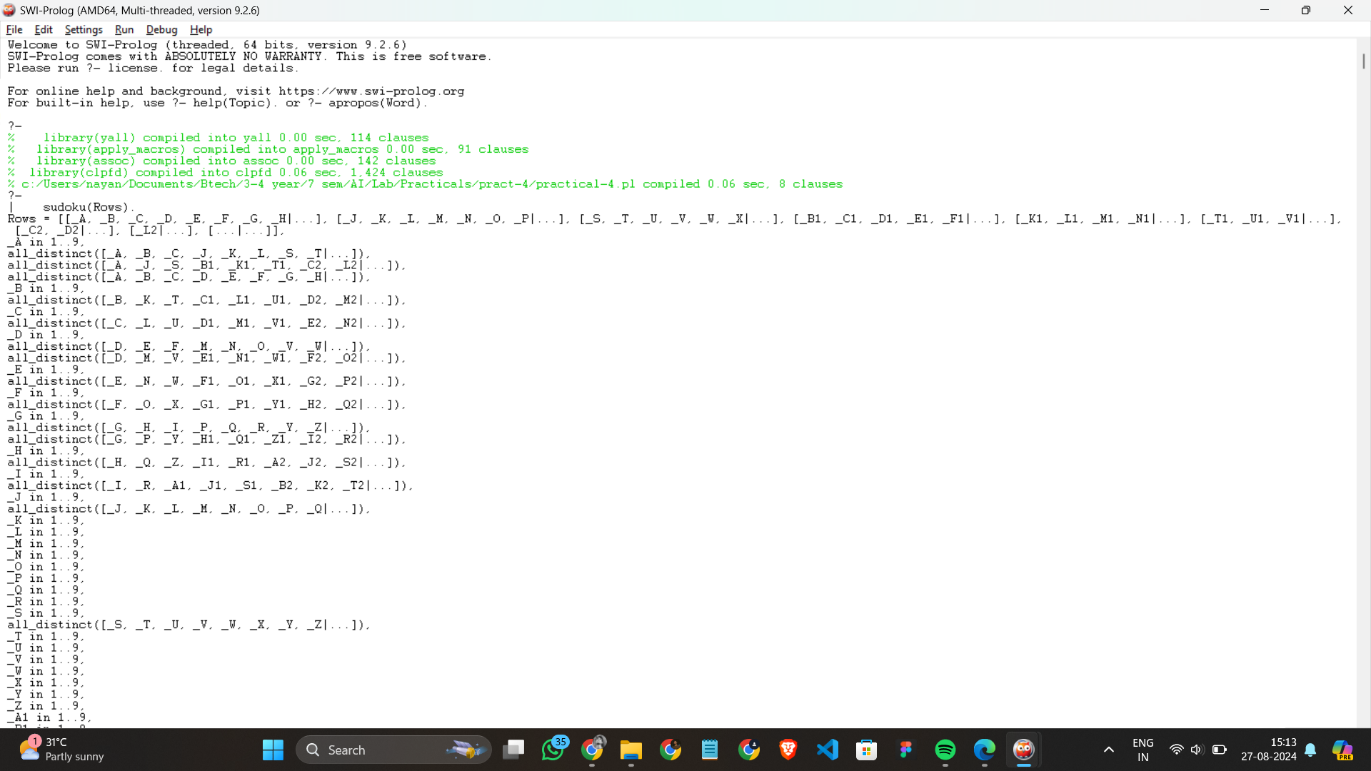
[\_,\_,\_,\_,\_,9,6,\_,\_],

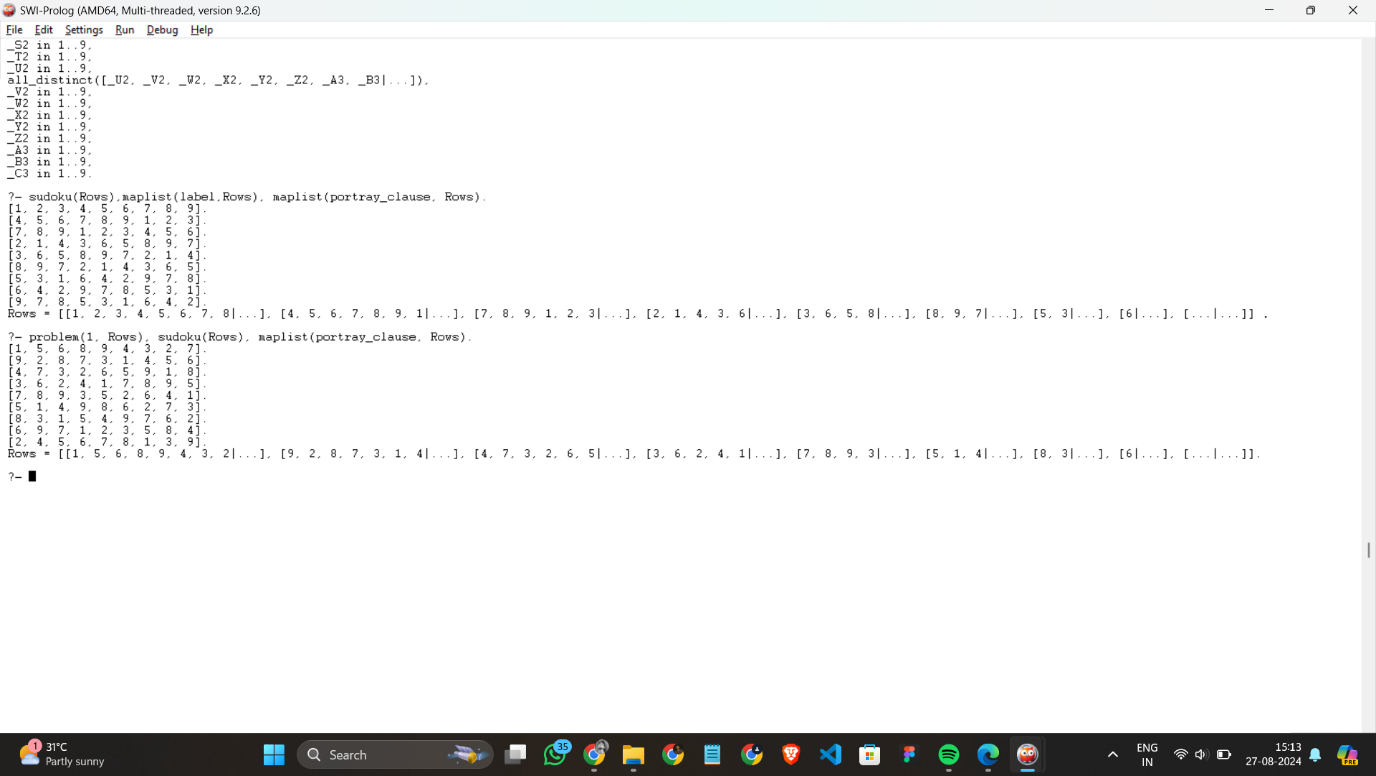
[\_,4,\_,\_,\_,6,\_,\_,\_],

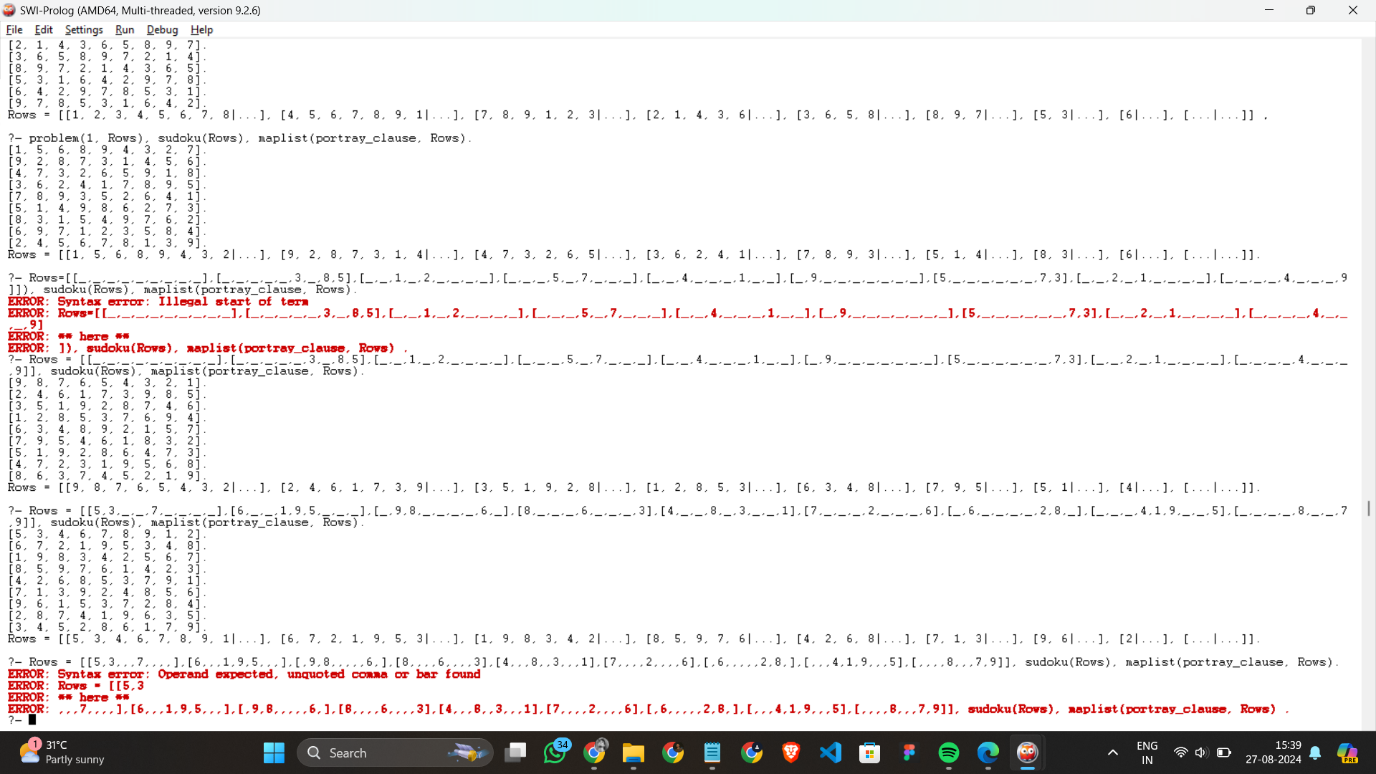
[\_,\_,\_,\_,\_,\_,\_,7,1],

[\_,\_,\_,\_,\_,1,\_,3,\_]].

**Output:**

****

****

****