**Artificial Intelligence Lab**

Code: PMDS601P

**Digital Assignment 4**

**&**

**Digital Assignment 5**

**Name: Soumyadeep Ganguly**

**Reg. No.: 24MDT0082**

**Course: M.Sc in Data Science**

**Digital Assignment 4**

1. Write a Prolog program defining a unary **predicate print\_upright\_tri\_ast/1** which takes as its argument a number N, and gives as output an up right triangle with N rows made up of asterisks. The first row in this triangle should have one star, the second should have two stars, etc., and the last row should have N stars.

**Code:**

**print\_ast(0):-**

write("").

**print\_ast(N):-**

N>0,

write("\*"),

X is N-1,

print\_ast(X).

**print\_upright\_tri\_ast(N):-**

**print\_tri\_helper(1,N).**

**print\_tri\_helper(Cur, N):-**

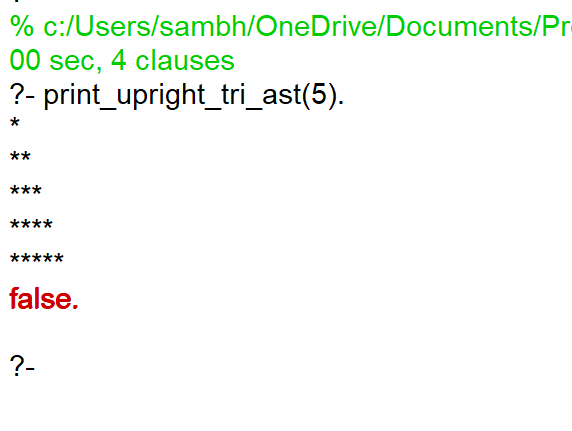
Cur =< N,

print\_ast(Cur),

nl,

Nxt is Cur + 1,

**print\_tri\_helper(Nxt, N).**

**Output:**

**Digital Assignment 5**

1. Write a Prolog program which prompts the user for 9 lists and checks whether the 9 ×9 grid forms a Latin square, that is, each column and each row contains all of the digits from 1 to 9. Further, you must also check that each of the nine 3 × 3 subgrids (blocks) that compose the.

**Code:**

element1([H|T],H).

element2([\_,H|T],H).

element3([\_,\_,H|T],H).

element4([\_,\_,\_,H|T],H).

element5([\_,\_,\_,\_,H|T],H).

element6([\_,\_,\_,\_,\_,H|T],H).

element7([\_,\_,\_,\_,\_,\_,H|T],H).

element8([\_,\_,\_,\_,\_,\_,\_,H|T],H).

element9([\_,\_,\_,\_,\_,\_,\_,\_,H|T],H).

**print\_grid :-**

write('Enter the 1st row :'), read(X1),

length(X1,9),

member(1,X1), member(2,X1), member(3,X1), member(4,X1), member(5,X1), member(6,X1), member(7,X1), member(8,X1), member(9,X1), nl,

write('Enter the 2nd row :'), read(X2),

length(X2,9),

member(1,X2), member(2,X2), member(3,X2), member(4,X2), member(5,X2), member(6,X2), member(7,X2), member(8,X2), member(9,X2), nl,

write('Enter the 3rd row :'), read(X3),

length(X3,9),

member(1,X3), member(2,X3), member(3,X3), member(4,X3), member(5,X3), member(6,X3), member(7,X3), member(8,X3), member(9,X3), nl,

write('Enter the 4th row :'), read(X4),

length(X4,9),

member(1,X4), member(2,X4), member(3,X4), member(4,X4), member(5,X4), member(6,X4), member(7,X4), member(8,X4), member(9,X4), nl,

write('Enter the 5th row :'), read(X5),

length(X5,9),

member(1,X5), member(2,X5), member(3,X5), member(4,X5), member(5,X5), member(6,X5), member(7,X5), member(8,X5), member(9,X5), nl,

write('Enter the 6th row :'), read(X6),

length(X6,9),

member(1,X6), member(2,X6), member(3,X6), member(4,X6), member(5,X6), member(6,X6), member(7,X6), member(8,X6), member(9,X6), nl,

write('Enter the 7th row :'), read(X7),

length(X7,9),

member(1,X7), member(2,X7), member(3,X7), member(4,X7), member(5,X7), member(6,X7), member(7,X7), member(8,X7), member(9,X7), nl,

write('Enter the 8th row :'), read(X8),

length(X8,9),

member(1,X8), member(2,X8), member(3,X8), member(4,X8), member(5,X8), member(6,X8), member(7,X8), member(8,X8), member(9,X8), nl,

write('Enter the 9th row :'), read(X9),

length(X9,9),

member(1,X9), member(2,X9), member(3,X9), member(4,X9), member(5,X9), member(6,X9), member(7,X9), member(8,X9), member(9,X9), nl,

element1(X1,A1), element1(X2,A2), element1(X3,A3), element1(X4,A4), element1(X5,A5), element1(X6,A6), element1(X7,A7), element1(X8,A8), element1(X9,A9),

C1 = [A1,A2,A3,A4,A5,A6,A7,A8,A9],

member(1,C1), member(2,C1), member(3,C1), member(4,C1), member(5,C1), member(6,C1), member(7,C1), member(8,C1), member(9,C1), nl,

element2(X1,B1), element2(X2,B2), element2(X3,B3), element2(X4,B4), element2(X5,B5), element2(X6,B6), element2(X7,B7), element2(X8,B8), element2(X9,B9),

C2 = [B1,B2,B3,B4,B5,B6,B7,B8,B9],

member(1,C2), member(2,C2), member(3,C2), member(4,C2), member(5,C2), member(6,C2), member(7,C2), member(8,C2), member(9,C2), nl,

element3(X1,D1), element3(X2,D2), element3(X3,D3), element3(X4,D4), element3(X5,D5), element3(X6,D6), element3(X7,D7), element3(X8,D8), element3(X9,D9),

C3 = [D1,D2,D3,D4,D5,D6,D7,D8,D9],

member(1,C3), member(2,C3), member(3,C3), member(4,C3), member(5,C3), member(6,C3), member(7,C3), member(8,C3), member(9,C3), nl,

element4(X1,E1), element4(X2,E2), element4(X3,E3), element4(X4,E4), element4(X5,E5), element4(X6,E6), element4(X7,E7), element4(X8,E8), element4(X9,E9),

C4 = [E1,E2,E3,E4,E5,E6,E7,E8,E9],

member(1,C4), member(2,C4), member(3,C4), member(4,C4), member(5,C4), member(6,C4), member(7,C4), member(8,C4), member(9,C4), nl,

element5(X1,F1), element5(X2,F2), element5(X3,F3), element5(X4,F4), element5(X5,F5), element5(X6,F6), element5(X7,F7), element5(X8,F8), element5(X9,F9),

C5 = [F1,F2,F3,F4,F5,F6,F7,F8,F9],

member(1,C5), member(2,C5), member(3,C5), member(4,C5), member(5,C5), member(6,C5), member(7,C5), member(8,C5), member(9,C5), nl,

element6(X1,G1), element6(X2,G2), element6(X3,G3), element6(X4,G4), element6(X5,G5), element6(X6,G6), element6(X7,G7), element6(X8,G8), element6(X9,G9),

C6 = [G1,G2,G3,G4,G5,G6,G7,G8,G9],

member(1,C6), member(2,C6), member(3,C6), member(4,C6), member(5,C6), member(6,C6), member(7,C6), member(8,C6), member(9,C6), nl,

element7(X1,H1), element7(X2,H2), element7(X3,H3), element7(X4,H4), element7(X5,H5), element7(X6,H6), element7(X7,H7), element7(X8,H8), element7(X9,H9),

C7 = [H1,H2,H3,H4,H5,H6,H7,H8,H9],

member(1,C7), member(2,C7), member(3,C7), member(4,C7), member(5,C7), member(6,C7), member(7,C7), member(8,C7), member(9,C7), nl,

element8(X1,I1), element8(X2,I2), element8(X3,I3), element8(X4,I4), element8(X5,I5), element8(X6,I6), element8(X7,I7), element8(X8,I8), element8(X9,I9),

C8 = [I1,I2,I3,I4,I5,I6,I7,I8,I9],

member(1,C8), member(2,C8), member(3,C8), member(4,C8), member(5,C8), member(6,C8), member(7,C8), member(8,C8), member(9,C8), nl,

element9(X1,J1), element9(X2,J2), element9(X3,J3), element9(X4,J4), element9(X5,J5), element9(X6,J6), element9(X7,J7), element9(X8,J8), element9(X9,J9),

C9 = [J1,J2,J3,J4,J5,J6,J7,J8,J9],

member(1,C9), member(2,C9), member(3,C9), member(4,C9), member(5,C9), member(6,C9), member(7,C9), member(8,C9), member(9,C9), nl,

element1(X1,SG11), element2(X1,SG12), element3(X1,SG13),

element1(X2,SG14), element2(X2,SG15), element3(X2,SG16),

element1(X3,SG17), element2(X3,SG18), element3(X3,SG19),

SG1 = [SG11,SG12,SG13,SG14,SG15,SG16,SG17,SG18,SG19],

member(1,SG1),member(2,SG1),member(3,SG1),member(4,SG1),member(5,SG1),member(6,SG1),member(7,SG1),member(8,SG1),member(9,SG1),

element4(X1,SG21), element5(X1,SG22), element6(X1,SG23),

element4(X2,SG24), element5(X2,SG25), element6(X2,SG26),

element4(X3,SG27), element5(X3,SG28), element6(X3,SG29),

SG2 = [SG21,SG22,SG23,SG24,SG25,SG26,SG27,SG28,SG29],

member(1,SG2),member(2,SG2),member(3,SG2),member(4,SG2),member(5,SG2),member(6,SG2),member(7,SG2),member(8,SG2),member(9,SG2),

element7(X1,SG31), element8(X1,SG32), element9(X1,SG33),

element7(X2,SG34), element8(X2,SG35), element9(X2,SG36),

element7(X3,SG37), element8(X3,SG38), element9(X3,SG39),

SG3 = [SG31,SG32,SG33,SG34,SG35,SG36,SG37,SG38,SG39],

member(1,SG3),member(2,SG3),member(3,SG3),member(4,SG3),member(5,SG3),member(6,SG3),member(7,SG3),member(8,SG3),member(9,SG3),

element1(X4,SG41), element2(X4,SG42), element3(X4,SG43),

element1(X5,SG44), element2(X5,SG45), element3(X5,SG46),

element1(X6,SG47), element2(X6,SG48), element3(X6,SG49),

SG4 = [SG41,SG42,SG43,SG44,SG45,SG46,SG47,SG48,SG49],

member(1,SG4),member(2,SG4),member(3,SG4),member(4,SG4),member(5,SG4),member(6,SG4),member(7,SG4),member(8,SG4),member(9,SG4),

element4(X4,SG51), element5(X4,SG52), element6(X4,SG53),

element4(X5,SG54), element5(X5,SG55), element6(X5,SG56),

element4(X6,SG57), element5(X6,SG58), element6(X6,SG59),

SG5 = [SG51,SG52,SG53,SG54,SG55,SG56,SG57,SG58,SG59],

member(1,SG5),member(2,SG5),member(3,SG5),member(4,SG5),member(5,SG5),member(6,SG5),member(7,SG5),member(8,SG5),member(9,SG5),

element7(X4,SG61), element8(X4,SG62), element9(X4,SG63),

element7(X5,SG64), element8(X5,SG65), element9(X5,SG66),

element7(X6,SG67), element8(X6,SG68), element9(X6,SG69),

SG6 = [SG61,SG62,SG63,SG64,SG65,SG66,SG67,SG68,SG69],

member(1,SG6),member(2,SG6),member(3,SG6),member(4,SG6),member(5,SG6),member(6,SG6),member(7,SG6),member(8,SG6),member(9,SG6),

element1(X7,SG71), element2(X7,SG72), element3(X7,SG73),

element1(X8,SG74), element2(X8,SG75), element3(X8,SG76),

element1(X9,SG77), element2(X9,SG78), element3(X9,SG79),

SG7 = [SG71,SG72,SG73,SG74,SG75,SG76,SG77,SG78,SG79],

member(1,SG7),member(2,SG7),member(3,SG7),member(4,SG7),member(5,SG7),member(6,SG7),member(7,SG7),member(8,SG7),member(9,SG7),

element4(X7,SG81), element5(X7,SG82), element6(X7,SG83), element4(X8,SG84), element5(X8,SG85), element6(X8,SG86),

element4(X9,SG87), element5(X9,SG88), element6(X9,SG89),

SG8 = [SG81,SG82,SG83,SG84,SG85,SG86,SG87,SG88,SG89],

member(1,SG8),member(2,SG8),member(3,SG8),member(4,SG8),member(5,SG8),member(6,SG8),member(7,SG8),member(8,SG8),member(9,SG8),

element7(X7,SG91), element8(X7,SG92), element9(X7,SG93),

element7(X8,SG94), element8(X8,SG95), element9(X8,SG96),

element7(X9,SG97), element8(X9,SG98), element9(X9,SG99),

SG9 = [SG91,SG92,SG93,SG94,SG95,SG96,SG97,SG98,SG99],

member(1,SG9),member(2,SG9),member(3,SG9),member(4,SG9),member(5,SG9),member(6,SG9),member(7,SG9),member(8,SG9),member(9,SG9),

write(X1), nl,

write(X2), nl,

write(X3), nl,

write(X4), nl,

write(X5), nl,

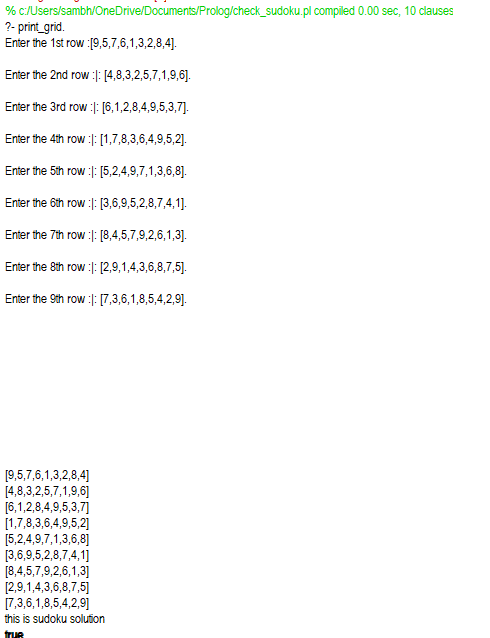
write(X6), nl,

write(X7), nl,

write(X8), nl,

write(X9), nl,

write('this is sudoku solution').

**Output:**