**CSE 579**

**Programming Assignment 1**

**Template for clingo Work**

Problem 1

|  |  |
| --- | --- |
| Input  Program | % Define the size of the chessboard  size(8).  % Generate positions for queens ensuring they are not in the middle 4x4 square  { queen(X,Y) : X = 1..8, Y = 1..8, not middle(X,Y) } = 8.  % Define the middle 4x4 squares  middle(3..6,3..6).  % Constraint: No two queens on the same row  :- queen(X,Y1), queen(X,Y2), Y1 != Y2.  % Constraint: No two queens on the same column  :- queen(X1,Y), queen(X2,Y), X1 != X2.  % Constraint: No two queens on the same diagonal  :- queen(X1,Y1), queen(X2,Y2), X1 != X2, abs(X1-X2) = abs(Y1-Y2).  % Display the solution.  #show queen/2. |
| Command  Line | clingo ./program\_1.txt 0 |
| Output  of clingo (Truncated) | queen(2,3) queen(5,7) queen(1,4) queen(4,8) queen(7,6) queen(3,1) queen(8,5) queen(6,2)  Answer: 16  queen(5,7) queen(2,5) queen(4,8) queen(1,6) queen(3,1) queen(7,4) queen(6,2) queen(8,3)  Answer: 17  queen(2,3) queen(6,8) queen(4,7) queen(1,5) queen(3,2) queen(8,6) queen(7,4) queen(5,1)  Answer: 18  queen(6,7) queen(1,3) queen(2,5) queen(4,8) queen(3,2) queen(8,6) queen(7,4) queen(5,1)  Answer: 19  queen(6,7) queen(2,5) queen(4,8) queen(1,6) queen(3,2) queen(7,4) queen(5,1) queen(8,3)  Answer: 20  queen(6,7) queen(2,4) queen(5,8) queen(1,5) queen(3,2) queen(8,6) queen(4,1) queen(7,3)  Answer: 21  queen(2,4) queen(4,7) queen(1,5) queen(3,8) queen(8,6) queen(5,2) queen(7,3) queen(6,1)  Answer: 22  queen(2,3) queen(6,8) queen(4,7) queen(1,5) queen(7,6) queen(3,1) queen(5,2) queen(8,4)  Answer: 23  queen(2,3) queen(4,7) queen(1,5) queen(3,8) queen(7,6) queen(5,2) queen(8,4) queen(6,1)  Answer: 24  queen(6,7) queen(1,3) queen(2,5) queen(4,8) queen(7,6) queen(3,1) queen(5,2) queen(8,4)  SATISFIABLE  Models : 24  Calls : 1  Time : 0.061s (Solving: 0.06s 1st Model: 0.00s Unsat: 0.03s)  CPU Time : 0.061s |

Problem 2

|  |  |
| --- | --- |
| Input  Program | #const n = 10.  % Define the svs of the chessboard in which the Queens can be placed.  {queen\_sv(R, 1..n)} = 1 :- R = 1..n.  % 2 queens not in the similar column  :- queen\_sv(R1, C), queen\_sv(R2, C), R1 != R2.  % queens not in diagonal  :- queen\_sv(R1, C1), queen\_sv(R2, C2), R1 != R2, |R1 - R2| = |C1 - C2|. |
| Command  Line | clingo -c n=3 p2.txt 0  clingo -c n=4 p2.txt 0  clingo -c n=5 p2.txt 0  clingo -c n=6 p2.txt 0  clingo -c n=7 p2.txt 0  clingo -c n=8 p2.txt 0  clingo -c n=9 p2.txt 0  clingo -c n=10 p2.txt 0  clingo -c n=11 p2.txt 0  clingo -c n=12 p2.txt 0 |
| Output  of clingo | - |
| Answer  to Questions | Draw a table that lists the number of solutions and the times to compute all solutions. Use CPU time that clingo returns.   |  |  |  | | --- | --- | --- | | Value n | Number of solutions | time (in sec) | | 3 | 0 | 0.001 | | 4 | 2 | 0.001 | | 5 | 10 | 0.001 | | 6 | 4 | 0.001 | | 7 | 40 | 0.002 | | 8 | 92 | 0.004 | | 9 | 352 | 0.02 | | 10 | 724 | 0.1 | | 11 | 2680 | 0.79 | | 12 | 14200 | 9.25 | |

Problem 3

|  |  |
| --- | --- |
| Input  Program | % Define the Sudoku board.  1 {sv(R, C, V): V = 1..9} 1 :- R = 1..9, C = 1..9.  % Given initial values on the Sudoku board.  sv(1,1,8).  sv(2,3,7). sv(2,4,5). sv(2,9,9).  sv(3,2,3). sv(3,7,1). sv(3,8,8).  sv(4,2,6). sv(4,6,1). sv(4,8,5).  sv(5,3,9). sv(5,5,4).  sv(6,4,7). sv(6,5,5).  sv(7,3,2). sv(7,5,7). sv(7,9,4).  sv(8,6,3). sv(8,7,6). sv(8,8,1).  sv(9,7,8).  % No two svs on the same column can have the same value.  :- sv(R1, C, V), sv(R2, C, V), R1 != R2.  % No two svs on the same row can have the same value.  :- sv(R, C1, V), sv(R, C2 ,V), C1 != C2.  % No two sv in the same subgrid can have the same value.  :- sv(R1, C1, V), sv(R2, C2, V), (R1-1)/3 == (R2-1)/3, (C1-1)/3 == (C2-1)/3, R1 != R2, C1 != C2.  #show sv/3. |
| Command  Line | clingo ./program\_3.txt 0 |
| Output  of clingo | clingo version 5.4.0  Reading from ./problem\_3.txt  Solving...  Answer: 1  sv(1,2,9) sv(1,7,4) sv(2,3,8) sv(2,4,5) sv(2,8,1) sv(3,3,1) sv(3,8,6) sv(3,9,8) sv(4,4,1) sv(4,8,3) sv(5,5,4) sv(5,6,5) sv(5,7,7) sv(6,2,5) sv(6,6,7) sv(7,2,7) sv(7,5,9) sv(7,7,2) sv(8,3,3) sv(8,4,6) sv(9,1,8) sv(3,2,2) sv(2,2,3) sv(1,1,7) sv(2,1,4) sv(3,1,5) sv(1,3,6) sv(2,5,2) sv(1,4,3) sv(3,6,4) sv(2,6,6) sv(3,5,7) sv(1,5,1) sv(3,4,9) sv(1,6,8) sv(2,9,7) sv(3,7,3) sv(1,8,5) sv(2,7,9) sv(1,9,2) sv(4,1,2) sv(6,3,4) sv(5,2,6) sv(5,3,9) sv(6,1,1) sv(5,1,3) sv(4,2,8) sv(4,3,7) sv(6,5,3) sv(6,4,2) sv(4,5,6) sv(5,4,8) sv(4,6,9) sv(5,9,1) sv(5,8,2) sv(4,7,5) sv(6,9,6) sv(6,8,9) sv(6,7,8) sv(4,9,4) sv(9,2,1) sv(9,3,2) sv(8,2,4) sv(7,1,6) sv(8,1,9) sv(7,3,5) sv(8,6,2) sv(9,6,3) sv(7,4,4) sv(9,5,5) sv(8,5,8) sv(9,4,7) sv(7,6,1) sv(9,8,4) sv(8,9,5) sv(8,8,7) sv(9,9,9) sv(8,7,1) sv(9,7,6) sv(7,8,8) sv(7,9,3)  SATISFIABLE  Models : 1  Calls : 1  Time : 0.017s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.017s |

Problem 4

|  |  |
| --- | --- |
| Input  Program | {sv(X,Y,N): X=1..16, Y=1..16, X1<=X, X<=X1+3, Y1<=Y, Y<=Y1+3} = 1 :- N=1..16, X1 = 4\*(0..3)+1, Y1 = 4\*(0..3)+1.  :- sv(X,Y,N), sv(X,Y,N1), N1!=N.  :- sv(X,Y,N), sv(X,Y1,N), Y1!=Y.  :- sv(X,Y,N), sv(X1,Y,N), X1!=X.  %Instance  sv(1,1,9).  sv(1,2,14).  sv(1,6,3).  sv(1,8,5).  sv(1,9,15).  sv(1,11,2).  sv(1,15,7).  sv(1,16,1).  sv(2,1,6).  sv(2,2,12).  sv(2,6,14).  sv(2,11,10).  sv(2,15,5).  sv(2,16,11).  sv(3,1,4).  sv(3,4,7).  sv(3,5,6).  sv(3,8,13).  sv(3,9,16).  sv(3,12,1).  sv(3,13,2).  sv(3,16,9).  sv(4,2,15).  sv(4,3,16).  sv(4,5,9).  sv(4,6,7).  sv(4,11,11).  sv(4,12,6).  sv(4,14,3).  sv(4,15,14).  sv(5,2,7).  sv(5,3,15).  sv(5,14,2).  sv(5,15,16).  sv(6,1,5).  sv(6,3,13).  sv(6,5,14).  sv(6,7,15).  sv(6,10,10).  sv(6,12,3).  sv(6,14,1).  sv(6,16,8).  sv(7,2,8).  sv(7,4,10).  sv(7,6,9).  sv(7,7,4).  sv(7,8,11).  sv(7,9,13).  sv(7,10,6).  sv(7,11,15).  sv(7,13,14).  sv(7,15,3).  sv(8,1,16).  sv(8,5,5).  sv(8,7,3).  sv(8,10,14).  sv(8,12,9).  sv(8,16,6).  sv(9,1,15).  sv(9,5,16).  sv(9,7,10).  sv(9,10,9).  sv(9,12,13).  sv(9,16,14).  sv(10,2,9).  sv(10,4,6).  sv(10,6,5).  sv(10,7,13).  sv(10,8,3).  sv(10,9,1).  sv(10,10,15).  sv(10,11,4).  sv(10,13,7).  sv(10,15,12).  sv(11,1,2).  sv(11,3,8).  sv(11,5,15).  sv(11,7,14).  sv(11,10,16).  sv(11,12,12).  sv(11,14,5).  sv(11,16,13).  sv(12,2,13).  sv(12,3,12).  sv(12,14,9).  sv(12,15,11).  sv(13,2,5).  sv(13,3,3).  sv(13,5,2).  sv(13,6,16).  sv(13,11,13).  sv(13,12,10).  sv(13,14,12).  sv(13,15,9).  sv(14,1,8).  sv(14,4,4).  sv(14,5,12).  sv(14,8,1).  sv(14,9,6).  sv(14,12,7).  sv(14,13,15).  sv(14,16,3).  sv(15,1,10).  sv(15,2,1).  sv(15,6,15).  sv(15,11,16).  sv(15,15,6).  sv(15,16,2).  sv(16,1,11).  sv(16,2,2).  sv(16,6,8).  sv(16,8,14).  sv(16,9,3).  sv(16,11,1).  sv(16,15,10).  sv(16,16,7). |
| Command  Line | clingo ./program\_4.txt 0 |
| Output  of clingo | clingo version 5.4.0  Reading from ./problem\_4.txt  Solving...  Answer: 1  sv(1,1,9) sv(1,2,14) sv(1,6,3) sv(1,8,5) sv(1,9,15) sv(1,11,2) sv(1,15,7) sv(1,16,1) sv(2,1,6) sv(2,2,12) sv(2,6,14) sv(2,11,10) sv(2,15,5) sv(2,16,11) sv(3,1,4) sv(3,4,7) sv(3,5,6) sv(3,8,13) sv(3,9,16) sv(3,12,1) sv(3,13,2) sv(3,16,9) sv(4,2,15) sv(4,3,16) sv(4,5,9) sv(4,6,7) sv(4,11,11) sv(4,12,6) sv(4,14,3) sv(4,15,14) sv(5,2,7) sv(5,3,15) sv(5,14,2) sv(5,15,16) sv(6,1,5) sv(6,3,13) sv(6,5,14) sv(6,7,15) sv(6,10,10) sv(6,12,3) sv(6,14,1) sv(6,16,8) sv(7,2,8) sv(7,4,10) sv(7,6,9) sv(7,7,4) sv(7,8,11) sv(7,9,13) sv(7,10,6) sv(7,11,15) sv(7,13,14) sv(7,15,3) sv(8,1,16) sv(8,5,5) sv(8,7,3) sv(8,10,14) sv(8,12,9) sv(8,16,6) sv(9,1,15) sv(9,5,16) sv(9,7,10) sv(9,10,9) sv(9,12,13) sv(9,16,14) sv(10,2,9) sv(10,4,6) sv(10,6,5) sv(10,7,13) sv(10,8,3) sv(10,9,1) sv(10,10,15) sv(10,11,4) sv(10,13,7) sv(10,15,12) sv(11,1,2) sv(11,3,8) sv(11,5,15) sv(11,7,14) sv(11,10,16) sv(11,12,12) sv(11,14,5) sv(11,16,13) sv(12,2,13) sv(12,3,12) sv(12,14,9) sv(12,15,11) sv(13,2,5) sv(13,3,3) sv(13,5,2) sv(13,6,16) sv(13,11,13) sv(13,12,10) sv(13,14,12) sv(13,15,9) sv(14,1,8) sv(14,4,4) sv(14,5,12) sv(14,8,1) sv(14,9,6) sv(14,12,7) sv(14,13,15) sv(14,16,3) sv(15,1,10) sv(15,2,1) sv(15,6,15) sv(15,11,16) sv(15,15,6) sv(15,16,2) sv(16,1,11) sv(16,2,2) sv(16,6,8) sv(16,8,14) sv(16,9,3) sv(16,11,1) sv(16,15,10) sv(16,16,7) sv(12,1,1) sv(2,3,1) sv(8,4,1) sv(7,3,2) sv(4,4,2) sv(5,1,3) sv(9,2,3) sv(2,4,3) sv(11,2,4) sv(8,3,4) sv(3,3,5) sv(9,4,5) sv(6,2,6) sv(16,3,6) sv(13,1,7) sv(9,3,7) sv(1,4,8) sv(15,3,9) sv(6,4,9) sv(3,2,10) sv(10,3,10) sv(8,2,11) sv(1,3,11) sv(11,4,11) sv(7,1,12) sv(16,4,12) sv(4,1,13) sv(15,4,13) sv(10,1,14) sv(14,3,14) sv(5,4,14) sv(13,4,15) sv(14,2,16) sv(12,4,16) sv(7,5,1) sv(9,6,1) sv(4,7,1) sv(8,6,2) sv(2,7,2) sv(12,8,2) sv(15,5,3) sv(2,5,4) sv(12,6,4) sv(15,8,4) sv(14,7,5) sv(11,6,6) sv(5,7,6) sv(13,8,6) sv(12,5,7) sv(15,7,7) sv(8,8,7) sv(5,5,8) sv(12,7,8) sv(4,8,8) sv(16,7,9) sv(11,8,9) sv(1,5,10) sv(14,6,10) sv(5,8,10) sv(10,5,11) sv(3,6,11) sv(13,7,11) sv(6,6,12) sv(3,7,12) sv(9,8,12) sv(16,5,13) sv(5,6,13) sv(2,8,15) sv(1,7,16) sv(6,8,16) sv(5,10,1) sv(6,9,2) sv(14,10,2) sv(10,12,2) sv(3,10,3) sv(11,11,3) sv(5,9,4) sv(16,10,4) sv(1,12,4) sv(4,9,5) sv(12,10,5) sv(5,11,5) sv(15,12,5) sv(12,11,6) sv(11,9,7) sv(2,10,7) sv(6,11,7) sv(8,9,8) sv(13,10,8) sv(9,11,8) sv(2,12,8) sv(2,9,9) sv(14,11,9) sv(12,9,10) sv(9,9,11) sv(15,10,11) sv(5,12,11) sv(15,9,12) sv(4,10,12) sv(8,11,12) sv(1,10,13) sv(13,9,14) sv(3,11,14) sv(12,12,14) sv(16,12,15) sv(7,12,16) sv(13,13,1) sv(11,15,1) sv(9,15,2) sv(12,13,3) sv(4,13,4) sv(9,14,4) sv(6,15,4) sv(13,16,4) sv(16,13,5) sv(7,16,5) sv(9,13,6) sv(1,14,6) sv(7,14,7) sv(15,13,8) sv(10,14,8) sv(3,15,8) sv(5,13,9) sv(11,13,10) sv(8,14,10) sv(4,16,10) sv(6,13,11) sv(14,14,11) sv(1,13,12) sv(5,16,12) sv(8,13,13) sv(2,14,13) sv(14,15,13) sv(15,14,14) sv(3,14,15) sv(8,15,15) sv(12,16,15) sv(2,13,16) sv(16,14,16) sv(10,16,16)  SATISFIABLE  Models : 1  Calls : 1  Time : 0.103s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.103s |

Problem 5

|  |  |
| --- | --- |
| Input  Program | % Define the Sudoku board.  1 {sv(R, C, V): V = 1..9} 1 :- R = 1..9, C = 1..9.  % Given initial values on the Sudoku board.  sv(1,3,6). sv(1, 7, 9).  sv(2,2,7). sv(2, 8, 8).  sv(3,1,3). sv(3, 3, 2). sv(3, 5, 8). sv(3, 7, 7). sv(3, 9, 4).  sv(4, 5, 9).  sv(5, 3, 8). sv(5, 4, 3). sv(5, 5, 6). sv(5, 6, 4). sv(5, 7, 2).  sv(6, 5, 7).  sv(7, 1, 8). sv(7, 3, 4). sv(7, 5, 2). sv(7, 7, 5). sv(7, 9, 1).  sv(8, 2, 2). sv(8, 8, 4).  sv(9, 3, 7). sv(9, 7, 8).  % No two svs on the same column can have the same value.  :- sv(R1, C, V), sv(R2, C, V), R1 != R2.  % No two svs on the same row can have the same value.  :- sv(R, C1, V), sv(R, C2, V), C1 != C2.  % No two sv in the same subgrid can have the same value.  :- sv(R1, C1, V), sv(R2, C2, V), (R1-1)/3 == (R2-1)/3, (C1-1)/3 == (C2-1)/3, R1 != R2, C1 != C2.  % No two svs in same offset locations can have the same value.  :- sv(R1, C1, V), sv(R2, C2, V), (R1-1)\3 == (R2-1)\3, (C1-1)\3 == (C2-1)\3, R1 != R2, C1 != C2.  #show sv/3. |
| Command  Line | clingo ./program\_5.txt 0 |
| Output  of clingo | clingo version 5.4.0  Reading from ./problem\_5.txt  Solving...  Answer: 1  sv(1,3,6) sv(1,7,9) sv(2,2,7) sv(2,8,8) sv(3,1,3) sv(3,3,2) sv(3,5,8) sv(3,7,7) sv(3,9,4) sv(4,5,9) sv(5,3,8) sv(5,4,3) sv(5,5,6) sv(5,6,4) sv(5,7,2) sv(6,5,7) sv(7,1,8) sv(7,3,4) sv(7,5,2) sv(7,7,5) sv(7,9,1) sv(8,2,2) sv(8,8,4) sv(9,3,7) sv(9,7,8) sv(4,4,1) sv(4,7,3) sv(1,1,4) sv(7,4,7) sv(7,8,3) sv(4,8,7) sv(1,2,8) sv(4,6,2) sv(4,9,8) sv(7,6,9) sv(1,4,2) sv(4,1,6) sv(1,5,1) sv(4,2,4) sv(7,2,6) sv(4,3,5) sv(1,6,7) sv(1,8,5) sv(1,9,3) sv(2,1,5) sv(8,7,6) sv(8,4,8) sv(5,8,1) sv(8,5,5) sv(8,6,1) sv(5,9,5) sv(8,9,7) sv(2,3,9) sv(2,4,4) sv(5,1,7) sv(8,1,9) sv(2,5,3) sv(5,2,9) sv(8,3,3) sv(2,6,6) sv(2,7,1) sv(2,9,2) sv(6,7,4) sv(6,4,5) sv(9,4,6) sv(3,2,1) sv(9,8,2) sv(9,5,4) sv(6,8,9) sv(9,6,3) sv(6,9,6) sv(6,6,8) sv(9,9,9) sv(9,1,1) sv(6,1,2) sv(3,4,9) sv(6,2,3) sv(9,2,5) sv(6,3,1) sv(3,6,5) sv(3,8,6)  SATISFIABLE  Models : 1  Calls : 1  Time : 0.032s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.032s |

Problem 6

|  |  |
| --- | --- |
| Input  Program | {sv(X,Y,N): X=1..9, Y=1..9, X1<=X, X<=X1+2, Y1<=Y, Y<=Y1+2} = 1 :- N=1..9, X1 = 3\*(0..2)+1, Y1 = 3\*(0..2)+1.  :- sv(X,Y,N), sv(X,Y,N1), N1!=N.  :- sv(X,Y,N), sv(X,Y1,N), Y1!=Y.  :- sv(X,Y,N), sv(X1,Y,N), X1!=X.  :- sv(X,Y,N), sv(X1,Y1,N), |X1-X|+|Y1-Y|==3.  %Instance  sv(1,1,3). sv(1,9,4).  sv(2,4,6). sv(2,6,9).  sv(3,3,6). sv(3,7,9).  sv(4,2,8). sv(4,4,3). sv(4,6,2). sv(4,8,6).  sv(5,5,7).  sv(6,2,1). sv(6,4,8). sv(6,6,5). sv(6,8,7).  sv(7,3,7). sv(7,7,8).  sv(8,4,7). sv(8,6,8).  sv(9,1,9). sv(9,9,7). |
| Command  Line | clingo ./problem\_6.txt 0 |
| Output  of clingo | clingo version 5.4.0  Reading from ./problem\_6.txt  Solving...  Answer: 1  sv(1,1,3) sv(1,9,4) sv(2,4,6) sv(2,6,9) sv(3,3,6) sv(3,7,9) sv(4,2,8) sv(4,4,3) sv(4,6,2) sv(4,8,6) sv(5,5,7) sv(6,2,1) sv(6,4,8) sv(6,6,5) sv(6,8,7) sv(7,3,7) sv(7,7,8) sv(8,4,7) sv(8,6,8) sv(9,1,9) sv(9,9,7) sv(1,3,1) sv(3,6,1) sv(4,5,1) sv(1,5,2) sv(2,2,2) sv(6,1,2) sv(3,5,3) sv(5,3,3) sv(2,1,4) sv(3,4,4) sv(6,3,4) sv(2,3,5) sv(1,4,5) sv(5,2,5) sv(5,1,6) sv(3,2,7) sv(4,1,7) sv(1,6,7) sv(3,1,8) sv(2,5,8) sv(1,2,9) sv(4,3,9) sv(5,4,9) sv(2,9,1) sv(3,9,2) sv(5,7,2) sv(2,8,3) sv(5,6,4) sv(4,7,4) sv(3,8,5) sv(1,7,6) sv(6,5,6) sv(2,7,7) sv(1,8,8) sv(5,8,1) sv(6,7,3) sv(4,9,5) sv(5,9,8) sv(6,9,9) sv(7,1,1) sv(8,3,2) sv(7,4,2) sv(9,2,3) sv(7,2,4) sv(8,1,5) sv(7,5,5) sv(8,2,6) sv(9,3,8) sv(9,4,1) sv(8,7,1) sv(7,6,3) sv(9,5,4) sv(9,6,6) sv(8,5,9) sv(7,8,9) sv(9,8,2) sv(8,9,3) sv(8,8,4) sv(9,7,5) sv(7,9,6)  SATISFIABLE  Models : 1  Calls : 1  Time : 0.032s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.032s |

Problem 7

|  |  |
| --- | --- |
| Input  Program | {sv(X,Y,N): X=1..9, Y=1..9, X1<=X, X<=X1+2, Y1<=Y, Y<=Y1+2} = 1 :- N=1..9, X1 = 3\*(0..2)+1, Y1 = 3\*(0..2)+1.  :- sv(X,Y,N), sv(X,Y,N1), N1!=N.  :- sv(X,Y,N), sv(X,Y1,N), Y1!=Y.  :- sv(X,Y,N), sv(X1,Y,N), X1!=X.  :- sv(X,Y,N), sv(X1,Y1,N1), gt(X,Y,X1,Y1), N <= N1.  %Instance  gt(1,2,1,1). gt(1,3,1,2). gt(1,3,2,3). gt(1,4,1,5). gt(1,6,1,5). gt(1,6,2,6). gt(1,7,2,7). gt(1,8,1,7). gt(1,8,2,8). gt(1,9,1,8). gt(1,9,2,9).  gt(2,1,1,1). gt(2,2,1,2). gt(2,2,2,1). gt(2,2,2,3). gt(2,2,3,2). gt(2,3,3,3). gt(2,4,1,4). gt(2,4,3,4). gt(2,5,1,5). gt(2,5,2,4). gt(2,5,2,6). gt(2,5,3,5). gt(2,6,3,6). gt(2,8,2,7). gt(2,9,2,8). gt(2,9,3,9).  gt(3,1,2,1). gt(3,1,3,2). gt(3,3,3,2). gt(3,4,3,5). gt(3,5,3,6). gt(3,7,2,7). gt(3,7,3,8). gt(3,8,2,8). gt(3,9,3,8).  gt(4,1,4,2). gt(4,1,5,1). gt(4,3,4,2). gt(4,3,5,3). gt(4,5,4,4). gt(4,6,4,5). gt(4,6,5,6). gt(4,7,4,8). gt(4,9,4,8).  gt(5,2,4,2). gt(5,2,5,1). gt(5,2,5,3). gt(5,2,6,2). gt(5,4,4,4). gt(5,4,5,5). gt(5,4,6,4). gt(5,5,4,5). gt(5,5,6,5). gt(5,6,5,5). gt(5,7,4,7). gt(5,7,5,8). gt(5,8,4,8). gt(5,8,5,9). gt(5,9,4,9).  gt(6,1,5,1). gt(6,2,6,1). gt(6,2,6,3). gt(6,3,5,3). gt(6,5,6,4). gt(6,6,5,6). gt(6,6,6,5). gt(6,7,5,7). gt(6,8,5,8). gt(6,8,6,7). gt(6,8,6,9).  gt(7,1,7,2). gt(7,1,8,1). gt(7,3,7,2). gt(7,3,8,2). gt(7,4,7,5). gt(7,4,8,4). gt(7,6,7,5). gt(7,6,8,6). gt(7,7,8,7). gt(7,8,7,7). gt(7,8,7,9).  gt(8,1,8,2). gt(8,1,9,1). gt(8,2,7,2). gt(8,2,8,3). gt(8,5,7,5). gt(8,5,8,4). gt(8,5,8,6). gt(8,6,9,6). gt(8,7,9,7). gt(8,8,7,8). gt(8,8,8,6). gt(8,8,9,6). gt(8,9,7,9). gt(8,9,8,8). gt(8,9,9,9).  gt(9,2,8,2). gt(9,2,9,1). gt(9,2,9,3). gt(9,3,8,3). gt(9,4,8,4). gt(9,5,8,5). gt(9,5,9,4). gt(9,5,9,6). gt(9,8,9,7). gt(9,9,9,8). |
| Command  Line | clingo ./problem\_7.txt 0 |
| Output  of clingo | clingo version 5.4.0  Reading from ./problem\_7.txt  Solving...  Answer: 1  gt(1,2,1,1) gt(1,3,1,2) gt(1,3,2,3) gt(1,4,1,5) gt(1,6,1,5) gt(1,6,2,6) gt(1,7,2,7) gt(1,8,1,7) gt(1,8,2,8) gt(1,9,1,8) gt(1,9,2,9) gt(2,1,1,1) gt(2,2,1,2) gt(2,2,2,1) gt(2,2,2,3) gt(2,2,3,2) gt(2,3,3,3) gt(2,4,1,4) gt(2,4,3,4) gt(2,5,1,5) gt(2,5,2,4) gt(2,5,2,6) gt(2,5,3,5) gt(2,6,3,6) gt(2,8,2,7) gt(2,9,2,8) gt(2,9,3,9) gt(3,1,2,1) gt(3,1,3,2) gt(3,3,3,2) gt(3,4,3,5) gt(3,5,3,6) gt(3,7,2,7) gt(3,7,3,8) gt(3,8,2,8) gt(3,9,3,8) gt(4,1,4,2) gt(4,1,5,1) gt(4,3,4,2) gt(4,3,5,3) gt(4,5,4,4) gt(4,6,4,5) gt(4,6,5,6) gt(4,7,4,8) gt(4,9,4,8) gt(5,2,4,2) gt(5,2,5,1) gt(5,2,5,3) gt(5,2,6,2) gt(5,4,4,4) gt(5,4,5,5) gt(5,4,6,4) gt(5,5,4,5) gt(5,5,6,5) gt(5,6,5,5) gt(5,7,4,7) gt(5,7,5,8) gt(5,8,4,8) gt(5,8,5,9) gt(5,9,4,9) gt(6,1,5,1) gt(6,2,6,1) gt(6,2,6,3) gt(6,3,5,3) gt(6,5,6,4) gt(6,6,5,6) gt(6,6,6,5) gt(6,7,5,7) gt(6,8,5,8) gt(6,8,6,7) gt(6,8,6,9) gt(7,1,7,2) gt(7,1,8,1) gt(7,3,7,2) gt(7,3,8,2) gt(7,4,7,5) gt(7,4,8,4) gt(7,6,7,5) gt(7,6,8,6) gt(7,7,8,7) gt(7,8,7,7) gt(7,8,7,9) gt(8,1,8,2) gt(8,1,9,1) gt(8,2,7,2) gt(8,2,8,3) gt(8,5,7,5) gt(8,5,8,4) gt(8,5,8,6) gt(8,6,9,6) gt(8,7,9,7) gt(8,8,7,8) gt(8,8,8,6) gt(8,8,9,6) gt(8,9,7,9) gt(8,9,8,8) gt(8,9,9,9) gt(9,2,8,2) gt(9,2,9,1) gt(9,2,9,3) gt(9,3,8,3) gt(9,4,8,4) gt(9,5,8,5) gt(9,5,9,4) gt(9,5,9,6) gt(9,8,9,7) gt(9,9,9,8) sv(1,1,2) sv(1,2,3) sv(1,3,9) sv(2,3,6) sv(1,5,1) sv(1,4,5) sv(1,6,4) sv(2,6,3) sv(2,7,1) sv(1,7,6) sv(1,8,7) sv(2,8,2) sv(1,9,8) sv(2,9,5) sv(2,1,4) sv(2,2,7) sv(3,2,1) sv(3,3,5) sv(2,4,8) sv(3,4,7) sv(2,5,9) sv(3,5,6) sv(3,6,2) sv(3,9,4) sv(3,1,8) sv(3,7,9) sv(3,8,3) sv(4,2,6) sv(4,1,9) sv(5,1,1) sv(4,3,7) sv(5,3,2) sv(4,4,3) sv(4,5,4) sv(4,6,8) sv(5,6,6) sv(4,8,1) sv(4,7,5) sv(4,9,2) sv(5,2,8) sv(6,2,5) sv(5,4,9) sv(5,5,5) sv(6,4,1) sv(6,5,2) sv(5,7,7) sv(5,8,4) sv(5,9,3) sv(6,1,3) sv(6,3,4) sv(6,6,7) sv(6,7,8) sv(6,8,9) sv(6,9,6) sv(7,2,2) sv(7,1,7) sv(8,1,6) sv(7,3,8) sv(8,2,4) sv(7,5,3) sv(7,4,6) sv(8,4,2) sv(7,6,9) sv(8,6,5) sv(8,7,3) sv(7,7,4) sv(7,8,5) sv(7,9,1) sv(9,1,5) sv(8,3,1) sv(8,5,7) sv(9,6,1) sv(9,7,2) sv(8,8,8) sv(8,9,9) sv(9,9,7) sv(9,2,9) sv(9,3,3) sv(9,4,4) sv(9,5,8) sv(9,8,6)  SATISFIABLE  Models : 1  Calls : 1  Time : 0.168s (Solving: 0.14s 1st Model: 0.04s Unsat: 0.10s)  CPU Time : 0.168s |

Problem 8

|  |  |
| --- | --- |
| Input  Program | {a(X,Y)} :- X=1..n, Y=1..n.  :- a(X,Y), a(X1,Y1), X!=X1, |X-X1|=|Y-Y1|.  #maximize{1,X,Y: a(X,Y)}. |
| Command  Line | clingo p8.txt -c n=3 0  clingo p8.txt -c n=4 0  clingo p8.txt -c n=5 0  clingo p8.txt -c n=6 0  clingo p8.txt -c n=7 0  clingo p8.txt -c n=8 0 |
| Output  of clingo | Models : 5  Optimum : yes  Optimization : -4  Calls : 1  Time : 0.001s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.001s  Models : 7  Optimum : yes  Optimization : -6  Calls : 1  Time : 0.001s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.001s  Models : 9  Optimum : yes  Optimization : -8  Calls : 1  Time : 0.003s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.003s  Models : 11  Optimum : yes  Optimization : -10  Calls : 1  Time : 0.010s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.01s)  CPU Time : 0.010s  Models : 13  Optimum : yes  Optimization : -12  Calls : 1  Time : 0.483s (Solving: 0.48s 1st Model: 0.00s Unsat: 0.48s)  CPU Time : 0.483s  Models : 15  Optimum : yes  Optimization : -14  Calls : 1  Time : 13.327s (Solving: 13.32s 1st Model: 0.00s Unsat: 13.11s)  CPU Time : 13.326s |
| Answer  to Questions | Draw a table that lists the maximum value of bishops when the chessboard is n by n, where n is 3, 4, 5, 6, 7, 8. Infer the general function f(n) that returns the maximum value of bishops.   |  |  | | --- | --- | | Value n | f(n) | | 3 | 4 | | 4 | 6 | | 5 | 8 | | 6 | 10 | | 7 | 12 | | 8 | 14 |   f(n) = (n-1)\*2 |

Problem 9

|  |  |
| --- | --- |
| Input  Program | {a(X,Y)} :- X=1..n, Y=1..n.  :- a(X,Y), a(X1,Y1), X!=X1, |X-X1|=|Y-Y1|.  #maximize{1,X,Y: a(X,Y)}. |
| Command  Line | clingo p9.txt -c k=1 -c n=2  clingo p9.txt -c k=2 -c n=8  clingo p9.txt -c k=3 -c n=23  clingo p9.txt -c k=4 -c n=66 |
| Output  of clingo | clingo version 5.4.0  Reading from problem\_9.txt  Solving...  Answer: 1  in(1,1) in(2,1)  SATISFIABLE  Models : 1  Calls : 1  Time : 0.001s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.001s  clingo version 5.4.0  Reading from problem\_9.txt  Solving...  Answer: 1  in(1,1) in(2,1) in(3,2) in(4,1) in(5,2) in(6,2) in(7,2) in(8,1)  Answer: 2  in(3,1) in(1,2) in(2,2) in(4,2) in(5,1) in(6,1) in(7,1) in(8,2)  SATISFIABLE  Models : 2  Calls : 1  Time : 0.002s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.002s  Answer: 14  in(3,1) in(1,2) in(2,2) in(4,2) in(5,1) in(6,1) in(7,1) in(8,2) in(9,3) in(10,3) in(11,2) in(12,3) in(13,3) in(14,3) in(15,3) in(16,2) in(17,3) in(18,3) in(19,1) in(20,3) in(21,1) in(22,2) in(23,1)  Answer: 15  in(3,1) in(1,2) in(2,2) in(4,2) in(5,1) in(6,1) in(7,1) in(8,2) in(9,3) in(10,3) in(11,2) in(12,3) in(13,3) in(14,3) in(15,3) in(16,3) in(17,2) in(18,3) in(19,1) in(20,3) in(21,1) in(22,2) in(23,1)  Answer: 16  in(3,1) in(1,3) in(2,3) in(4,3) in(5,1) in(6,1) in(7,1) in(8,3) in(9,2) in(10,2) in(11,3) in(12,2) in(13,2) in(14,2) in(15,2) in(16,3) in(17,2) in(18,2) in(19,1) in(20,2) in(21,1) in(22,3) in(23,1)  Answer: 17  in(3,1) in(1,3) in(2,3) in(4,3) in(5,1) in(6,1) in(7,1) in(8,3) in(9,2) in(10,2) in(11,3) in(12,2) in(13,2) in(14,2) in(15,2) in(16,2) in(17,2) in(18,2) in(19,1) in(20,2) in(21,1) in(22,3) in(23,1)  Answer: 18  in(3,1) in(1,3) in(2,3) in(4,3) in(5,1) in(6,1) in(7,1) in(8,3) in(9,2) in(10,2) in(11,3) in(12,2) in(13,2) in(14,2) in(15,2) in(16,2) in(17,3) in(18,2) in(19,1) in(20,2) in(21,1) in(22,3) in(23,1)  SATISFIABLE  Models : 18  Calls : 1  Time : 0.027s (Solving: 0.02s 1st Model: 0.01s Unsat: 0.00s)  CPU Time : 0.027s |
| Answer  to Questions | Fill in the values accordingly.   |  |  | | --- | --- | | Exact value of A(1) | 1 | | Exact value of A(2) | 2 | | Exact value of A(3) | 18 | | Largest lower bound for A(4)  Note: it would take longer time when you increase the value of n. Thus, you may stop increasing the value of n when your program does not terminate within 10 minutes and submit the last trial of n. | Did not terminate after 5 minutes. | |