

Problem 1

Input Program	<pre>%Each row has exactly one queen {queen(R,1..8)}=1:- R=1..8. %No two queens are on the same column :- queen(R0,C), queen(R1,C), R0!=R1. %No two queens are on the same diagonal :- queen(R0,C0), queen(R1,C1), R0!=R1, R0- R1 = C0-C1 . %No queens in the middle 4x4 squares :- queen(3..6,3..6).</pre>
Command Line	clingo queen.txt
Output of clingo	<pre>Answer: 1 queen(5,7) queen(1,4) queen(2,6) queen(4,2) queen(3,8) queen(6,1) queen(7,3) queen(8,5) Answer: 2 queen(2,3) queen(3,1) queen(6,8) queen(4,7) queen(1,5) queen(5,2) queen(7,6) queen(8,4) Answer: 3 queen(2,4) queen(4,1) queen(5,8) queen(3,7) queen(1,6) queen(6,2) queen(7,5) queen(8,3) Answer: 4 queen(6,7) queen(1,3) queen(2,5) queen(3,2) queen(4,8) queen(5,1) queen(8,6) queen(7,4) SATISFIABLE Models : 4 Calls : 1 Time : 0.059s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s) CPU Time : 0.000s</pre>

Problem 2

Input Program	<pre>%Each row has exactly one queen {queen(R,1..n)}=1:- R=1..n. %No two queens are on the same column :- queen(R0,C), queen(R1,C), R0!=R1. %No two queens are on the same diagonal :- queen(R0,C0), queen(R1,C1), R0!=R1, R0- R1 = C0-C1 .</pre>																																			
Command Line	<pre>clingo nqueens.txt -c n=3 0 clingo nqueens.txt -c n=4 0 clingo nqueens.txt -c n=5 0 clingo nqueens.txt -c n=6 0 clingo nqueens.txt -c n=7 0 clingo nqueens.txt -c n=8 0 clingo nqueens.txt -c n=9 0 clingo nqueens.txt -c n=10 0 clingo nqueens.txt -c n=11 0 clingo nqueens.txt -c n=12 0</pre>																																			
Output of clingo	N.A.																																			
Answer to Questions	<p>Draw a table that lists the number of solutions and the times to compute all solutions. Use CPU time that clingo returns.</p> <table><tr><th>Value n</th><th>Number of solutions</th><th>time</th></tr><tr><td>3</td><td>0</td><td>0.000s</td></tr><tr><td>4</td><td>2</td><td>0.000s</td></tr><tr><td>5</td><td>10</td><td>0.000s</td></tr><tr><td>6</td><td>4</td><td>0.000s</td></tr><tr><td>7</td><td>40</td><td>0.031s</td></tr><tr><td>8</td><td>92</td><td>0.188s</td></tr><tr><td>9</td><td>352</td><td>0.547s</td></tr><tr><td>10</td><td>724</td><td>1.406s</td></tr><tr><td>11</td><td>2680</td><td>6.500s</td></tr><tr><td>12</td><td>14200</td><td>39.922s</td></tr></table>			Value n	Number of solutions	time	3	0	0.000s	4	2	0.000s	5	10	0.000s	6	4	0.000s	7	40	0.031s	8	92	0.188s	9	352	0.547s	10	724	1.406s	11	2680	6.500s	12	14200	39.922s
Value n	Number of solutions	time																																		
3	0	0.000s																																		
4	2	0.000s																																		
5	10	0.000s																																		
6	4	0.000s																																		
7	40	0.031s																																		
8	92	0.188s																																		
9	352	0.547s																																		
10	724	1.406s																																		
11	2680	6.500s																																		
12	14200	39.922s																																		

Problem 3

Input Program	<pre>% Each no 1 to 9 is assigned to one cell in individual box 1{sudo(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. % no two different numbers given a row and a column :- sudo(X,Y,N1), sudo(X,Y,N2), N1!=N2. % no two different columns given a row and a number :- sudo(X,Y1,N), sudo(X,Y2,N), Y1!=Y2. %no two different rows given a column and a number :- sudo(X1,Y,N), sudo(X2,Y,N), X1!=X2.</pre>
Command Line	clingo Sudoku.txt Sudoku-instance.txt 0
Output of clingo	<pre>Answer: 1 sudo(1,1,8) sudo(2,3,3) sudo(2,4,6) sudo(3,2,7) sudo(3,5,9) sudo(3,7,2) sudo(4,2,5) sudo(4,6,7) sudo(5,5,4) sudo(5,6,5) sudo(5,7,7) sudo(6,4,1) sudo(6,8,3) sudo(7,3,1) sudo(7,8,6) sudo(7,9,8) sudo(8,3,8) sudo(8,4,5) sudo(8,8,1) sudo(9,2,9) sudo(9,7,4) sudo(4,1,1) sudo(1,2,1) sudo(6,1,2) sudo(7,2,2) sudo(1,3,2) sudo(5,1,3) sudo(8,2,3) sudo(8,1,4) sudo(2,2,4) sudo(4,3,4) sudo(7,1,5) sudo(3,3,5) sudo(3,1,6) sudo(5,2,6) sudo(9,3,6) sudo(9,1,7) sudo(6,3,7) sudo(6,2,8) sudo(2,1,9) sudo(5,3,9) sudo(9,5,1) sudo(3,6,1) sudo(4,4,2) sudo(8,5,2) sudo(2,6,2) sudo(9,4,3) sudo(4,5,3) sudo(1,6,3) sudo(3,4,4) sudo(7,6,4) sudo(1,5,5) sudo(6,5,6) sudo(8,6,6) sudo(1,4,7) sudo(7,5,7) sudo(5,4,8) sudo(2,5,8) sudo(9,6,8) sudo(7,4,9) sudo(6,6,9) sudo(2,7,1) sudo(5,9,1) sudo(5,8,2) sudo(9,9,2) sudo(7,7,3) sudo(3,9,3) sudo(1,8,4) sudo(6,9,4) sudo(6,7,5) sudo(9,8,5) sudo(2,9,5) sudo(1,7,6) sudo(4,9,6) sudo(2,8,7) sudo(8,9,7) sudo(4,7,8) sudo(3,8,8) sudo(8,7,9) sudo(4,8,9) sudo(1,9,9) SATISFIABLE Models : 1 Calls : 1</pre>

	Time : 0.039s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.01s) CPU Time : 0.031s
--	--

Problem 4

<p>Input Program</p>	<pre>% Each no 1 to 16 is assigned to one cell in individual box 1{sudo(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. % no two different numbers given a row and a column :- sudo(X,Y,N1), sudo(X,Y,N2), N1!=N2. % no two different columns given a row and a number :- sudo(X,Y1,N), sudo(X,Y2,N), Y1!=Y2. %no two different rows given a column and a number :- sudo(X1,Y,N), sudo(X2,Y,N), X1!=X2.</pre>
<p>Command Line</p>	<pre>clingo Sudoku16.txt Sudoku16-input.txt 0</pre>
<p>Output of clingo</p>	<pre>Answer: 1 sudo(1,1,9) sudo(1,2,14) sudo(1,6,3) sudo(1,8,5) sudo(1,9,15) sudo(1,11,2) sudo(1,15,7) sudo(1,16,1) sudo(2,1,6) sudo(2,2,12) sudo(2,6,14) sudo(2,11,10) sudo(2,15,5) sudo(2,16,11) sudo(3,1,4) sudo(3,4,7) sudo(3,5,6) sudo(3,8,13) sudo(3,9,16) sudo(3,12,1) sudo(3,13,2) sudo(3,16,9) sudo(4,2,15) sudo(4,3,16) sudo(4,5,9) sudo(4,6,7) sudo(4,11,11) sudo(4,12,6) sudo(4,14,3) sudo(4,15,14) sudo(5,2,7) sudo(5,3,15) sudo(5,14,2) sudo(5,15,16) sudo(6,1,5) sudo(6,3,13) sudo(6,5,14) sudo(6,7,15) sudo(6,10,10) sudo(6,12,3) sudo(6,14,1) sudo(6,16,8) sudo(7,2,8) sudo(7,4,10) sudo(7,6,9) sudo(7,7,4) sudo(7,8,11) sudo(7,9,13) sudo(7,10,6) sudo(7,11,15) sudo(7,13,14) sudo(7,15,3) sudo(8,1,16) sudo(8,5,5) sudo(8,7,3) sudo(8,10,14) sudo(8,12,9) sudo(8,16,6) sudo(9,1,15) sudo(9,5,16) sudo(9,7,10) sudo(9,10,9) sudo(9,12,13) sudo(9,16,14) sudo(10,2,9) sudo(10,4,6) sudo(10,6,5) sudo(10,7,13) sudo(10,8,3) sudo(10,9,1) sudo(10,10,15) sudo(10,11,4) sudo(10,13,7) sudo(10,15,12) sudo(11,1,2) sudo(11,3,8) sudo(11,5,15) sudo(11,7,14) sudo(11,10,16) sudo(11,12,12) sudo(11,14,5) sudo(11,16,13) sudo(12,2,13) sudo(12,3,12) sudo(12,14,9) sudo(12,15,11)</pre>

<p> sudo(13,2,5) sudo(13,3,3) sudo(13,5,2) sudo(13,6,16) sudo(13,11,13) sudo(13,12,10) sudo(13,14,12) sudo(13,15,9) sudo(14,1,8) sudo(14,4,4) sudo(14,5,12) sudo(14,8,1) sudo(14,9,6) sudo(14,12,7) sudo(14,13,15) sudo(14,16,3) sudo(15,1,10) sudo(15,2,1) sudo(15,6,15) sudo(15,11,16) sudo(15,15,6) sudo(15,16,2) sudo(16,1,11) sudo(16,2,2) sudo(16,6,8) sudo(16,8,14) sudo(16,9,3) sudo(16,11,1) sudo(16,15,10) sudo(16,16,7) sudo(12,1,1) sudo(2,3,1) sudo(8,4,1) sudo(7,3,2) sudo(4,4,2) sudo(5,1,3) sudo(9,2,3) sudo(2,4,3) sudo(11,2,4) sudo(8,3,4) sudo(3,3,5) sudo(9,4,5) sudo(6,2,6) sudo(16,3,6) sudo(13,1,7) sudo(9,3,7) sudo(1,4,8) sudo(15,3,9) sudo(6,4,9) sudo(3,2,10) sudo(10,3,10) sudo(8,2,11) sudo(1,3,11) sudo(11,4,11) sudo(7,1,12) sudo(16,4,12) sudo(4,1,13) sudo(15,4,13) sudo(10,1,14) sudo(14,3,14) sudo(5,4,14) sudo(13,4,15) sudo(14,2,16) sudo(12,4,16) sudo(7,5,1) sudo(9,6,1) sudo(4,7,1) sudo(8,6,2) sudo(2,7,2) sudo(12,8,2) sudo(15,5,3) sudo(2,5,4) sudo(12,6,4) sudo(15,8,4) sudo(14,7,5) sudo(11,6,6) sudo(5,7,6) sudo(13,8,6) sudo(12,5,7) sudo(15,7,7) sudo(8,8,7) sudo(5,5,8) sudo(12,7,8) sudo(4,8,8) sudo(16,7,9) sudo(11,8,9) sudo(1,5,10) sudo(14,6,10) sudo(5,8,10) sudo(10,5,11) sudo(3,6,11) sudo(13,7,11) sudo(6,6,12) sudo(3,7,12) sudo(9,8,12) sudo(16,5,13) sudo(5,6,13) sudo(2,8,15) sudo(1,7,16) sudo(6,8,16) sudo(5,10,1) sudo(6,9,2) sudo(14,10,2) sudo(10,12,2) sudo(3,10,3) sudo(11,11,3) sudo(5,9,4) sudo(16,10,4) sudo(1,12,4) sudo(4,9,5) sudo(12,10,5) sudo(5,11,5) sudo(15,12,5) sudo(12,11,6) sudo(11,9,7) sudo(2,10,7) sudo(6,11,7) sudo(8,9,8) sudo(13,10,8) sudo(9,11,8) sudo(2,12,8) sudo(2,9,9) sudo(14,11,9) sudo(12,9,10) sudo(9,9,11) sudo(15,10,11) sudo(5,12,11) sudo(15,9,12) sudo(4,10,12) sudo(8,11,12) sudo(1,10,13) sudo(13,9,14) sudo(3,11,14) sudo(12,12,14) sudo(16,12,15) sudo(7,12,16) sudo(13,13,1) sudo(11,15,1) sudo(9,15,2) sudo(12,13,3) sudo(4,13,4) sudo(9,14,4) sudo(6,15,4) sudo(13,16,4) sudo(16,13,5) sudo(7,16,5) sudo(9,13,6) sudo(1,14,6) sudo(7,14,7) sudo(15,13,8) sudo(10,14,8) sudo(3,15,8) </p>
--

	<pre> sudo(5,13,9) sudo(11,13,10) sudo(8,14,10) sudo(4,16,10) sudo(6,13,11) sudo(14,14,11) sudo(1,13,12) sudo(5,16,12) sudo(8,13,13) sudo(2,14,13) sudo(14,15,13) sudo(15,14,14) sudo(3,14,15) sudo(8,15,15) sudo(12,16,15) sudo(2,13,16) sudo(16,14,16) sudo(10,16,16) SATISFIABLE Models : 1 Calls : 1 Time : 0.199s (Solving: 0.05s 1st Model: 0.00s Unsat: 0.05s) CPU Time : 0.156s </pre>
--	---

Problem 5

<p>Input Program</p>	<pre>% Each no 1 to 9 is assigned to one cell in individual box {sudo(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. % no two different columns given a row and a number :- sudo(X,Y1,N), sudo(X,Y2,N), Y1!=Y2. %no two different rows given a column and a number :- sudo(X1,Y,N), sudo(X2,Y,N), X1!=X2. % no two different numbers given a row and a column :- sudo(X,Y,N1), sudo(X,Y,N2), N1!=N2. %offset condition :- sudo(X1,Y1,N), sudo(X2,Y2,N), X1\3==X2\3, Y1\3==Y2\3, X1!=X2, Y1!=Y2.</pre>
<p>Command Line</p>	<pre>clingo SudokuOffset.txt SudokuOffset-input.txt 0</pre>
<p>Output of clingo</p>	<pre>Answer: 1 sudo(1,3,7) sudo(1,7,8) sudo(2,2,2) sudo(2,8,4) sudo(3,1,8) sudo(3,3,4) sudo(3,5,2) sudo(3,7,5) sudo(3,9,1) sudo(4,5,7) sudo(5,3,8) sudo(5,4,3) sudo(5,5,6) sudo(5,6,4) sudo(5,7,2) sudo(6,5,9) sudo(7,1,3) sudo(7,3,2) sudo(7,5,8) sudo(7,7,7) sudo(7,9,4) sudo(8,2,7) sudo(8,8,8) sudo(9,3,6) sudo(9,7,9) sudo(1,1,1) sudo(8,7,1) sudo(5,8,1) sudo(6,4,1) sudo(9,5,1) sudo(8,9,2) sudo(9,4,2) sudo(6,6,2) sudo(8,5,3) sudo(2,3,3) sudo(6,7,3) sudo(9,9,3) sudo(4,7,4) sudo(8,4,4) sudo(4,4,5) sudo(1,2,5) sudo(7,6,5) sudo(5,9,5) sudo(9,8,5) sudo(7,8,6) sudo(4,9,6) sudo(8,6,6) sudo(3,2,6) sudo(6,8,7) sudo(9,6,7) sudo(4,6,8) sudo(6,9,8) sudo(7,4,9) sudo(4,8,9) sudo(2,1,9) sudo(7,2,1) sudo(4,3,1) sudo(2,6,1) sudo(4,1,2) sudo(4,2,3) sudo(1,6,3) sudo(1,5,4) sudo(9,1,4) sudo(6,2,4) sudo(8,1,5) sudo(2,5,5) sudo(6,3,5) sudo(1,4,6) sudo(6,1,6) sudo(5,1,7) sudo(3,4,7) sudo(2,4,8) sudo(9,2,8) sudo(5,2,9) sudo(8,3,9) sudo(3,6,9) sudo(1,8,2) sudo(3,8,3) sudo(2,7,6) sudo(2,9,7) sudo(1,9,9) SATISFIABLE</pre>

	Models : 1 Calls : 1 Time : 0.055s (Solving: 0.02s 1st Model: 0.00s Unsat: 0.02s) CPU Time : 0.031s
--	---

Problem 6

<p>Input Program</p>	<pre>% Each no 1 to 9 is assigned to one cell in individual box {sudo(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. % no two different columns given a row and a number :- sudo(X,Y1,N), sudo(X,Y2,N), Y1!=Y2. %no two different rows given a column and a number :- sudo(X1,Y,N), sudo(X2,Y,N), X1!=X2. % no two different numbers given a row and a column :- sudo(X,Y,N1), sudo(X,Y,N2), N1!=N2. %anti-knight condition :- sudo(X1,Y1,N), sudo(X2,Y2,N), X1-X2 + Y1- Y2 ==3.</pre>
<p>Command Line</p>	<pre>clingo SudokuKnight.txt SudokuKnight_input.txt 0</pre>
<p>Output of clingo</p>	<pre>Answer: 1 sudo(1,1,3) sudo(1,9,4) sudo(2,4,6) sudo(2,6,9) sudo(3,3,6) sudo(3,7,9) sudo(4,2,8) sudo(4,4,3) sudo(4,6,2) sudo(4,8,6) sudo(5,5,7) sudo(6,2,1) sudo(6,4,8) sudo(6,6,5) sudo(6,8,7) sudo(7,3,7) sudo(7,7,8) sudo(8,4,7) sudo(8,6,8) sudo(9,1,9) sudo(9,9,7) sudo(1,3,1) sudo(3,6,1) sudo(4,5,1) sudo(1,5,2) sudo(2,2,2) sudo(6,1,2) sudo(3,5,3) sudo(5,3,3) sudo(2,1,4) sudo(3,4,4) sudo(6,3,4) sudo(2,3,5) sudo(1,4,5) sudo(5,2,5) sudo(5,1,6) sudo(3,2,7) sudo(4,1,7) sudo(1,6,7) sudo(3,1,8) sudo(2,5,8) sudo(1,2,9) sudo(4,3,9) sudo(5,4,9) sudo(2,9,1) sudo(3,9,2) sudo(5,7,2) sudo(2,8,3) sudo(5,6,4) sudo(4,7,4) sudo(3,8,5) sudo(1,7,6) sudo(6,5,6) sudo(2,7,7) sudo(1,8,8) sudo(5,8,1) sudo(6,7,3) sudo(4,9,5) sudo(5,9,8) sudo(6,9,9) sudo(7,1,1) sudo(8,3,2) sudo(7,4,2) sudo(9,2,3) sudo(7,2,4) sudo(8,1,5) sudo(7,5,5) sudo(8,2,6) sudo(9,3,8) sudo(9,4,1) sudo(8,7,1) sudo(7,6,3) sudo(9,5,4) sudo(9,6,6) sudo(8,5,9) sudo(7,8,9) sudo(9,8,2) sudo(8,9,3) sudo(8,8,4) sudo(9,7,5) sudo(7,9,6) SATISFIABLE</pre>

	Models : 1
	Calls : 1
	Time : 0.061s (Solving: 0.02s 1st Model: 0.00s Unsat: 0.02s)
	CPU Time : 0.047s

Problem 7

Input Program	<pre>% Each no 1 to 9 is assigned to one cell in individual box 1{sudo(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. % no two different numbers given a row and a column :- sudo(X,Y,N1), sudo(X,Y,N2), N1!=N2. % no two different columns given a row and a number :- sudo(X,Y1,N), sudo(X,Y2,N), Y1!=Y2. %no two different rows given a column and a number :- sudo(X1,Y,N), sudo(X2,Y,N), X1!=X2. %GreaterThan Condition :- sudo(X1,Y1,N1), sudo(X2,Y2,N2), greaterthan(X1,Y1,X2,Y2), N1<=N2. #show sudo/3.</pre>
Command Line	clingo GTSudoku.txt GTSudoku-input.txt 0
Output of clingo	<pre>Answer: 1 sudo(1,2,7) sudo(1,1,8) sudo(1,3,1) sudo(2,1,6) sudo(3,1,2) sudo(3,2,9) sudo(3,3,5) sudo(2,3,4) sudo(2,2,3) sudo(1,4,5) sudo(1,5,9) sudo(1,6,6) sudo(2,4,2) sudo(3,4,3) sudo(3,5,4) sudo(3,6,8) sudo(2,6,7) sudo(2,5,1) sudo(1,8,3) sudo(1,7,4) sudo(1,9,2) sudo(2,7,9) sudo(3,7,1) sudo(3,8,7) sudo(3,9,6) sudo(2,9,5) sudo(2,8,8) sudo(4,1,1) sudo(4,2,4) sudo(4,3,3) sudo(5,1,9) sudo(6,1,7) sudo(6,2,5) sudo(6,3,6) sudo(5,3,8) sudo(5,2,2) sudo(4,5,6) sudo(4,4,7) sudo(4,6,2) sudo(5,4,1) sudo(6,4,9) sudo(6,5,8) sudo(6,6,3) sudo(5,6,4) sudo(5,5,5) sudo(4,7,5) sudo(4,8,9) sudo(4,9,8) sudo(5,7,3) sudo(6,7,2) sudo(6,8,1) sudo(6,9,4) sudo(5,9,7) sudo(5,8,6) sudo(7,1,3) sudo(7,2,8) sudo(7,3,2) sudo(8,1,4) sudo(9,1,5) sudo(9,2,1) sudo(9,3,7) sudo(8,3,9) sudo(8,2,6) sudo(7,4,4) sudo(7,5,7) sudo(7,6,1) sudo(8,4,8) sudo(9,4,6) sudo(9,5,2) sudo(9,6,9) sudo(8,6,5) sudo(8,5,3) sudo(7,8,5) sudo(7,7,6) sudo(7,9,9) sudo(8,7,7)</pre>

	sudo(9,7,8) sudo(9,8,4) sudo(9,9,3) sudo(8,9,1) sudo(8,8,2) SATISFIABLE Models : 1 Calls : 1 Time : 0.569s (Solving: 0.51s 1st Model: 0.33s Unsat: 0.19s) CPU Time : 0.547s
--	--

Problem 8

<p>Input Program</p>	<pre>%Bishop on each row {bishop(X,1..n)}:-X=1..n. %No two bishops on the same diagonal :- bishop(R1,C1), bishop(R2,C2), R1!=R2, R1-R2 = C1-C2 . %Maximize #maximize{1,X,Y : bishop(X,Y)}.</pre>
<p>Command Line</p>	<pre>clingo bishop.txt -c n=3 clingo bishop.txt -c n=4 clingo bishop.txt -c n=5 clingo bishop.txt -c n=6 clingo bishop.txt -c n=7 clingo bishop.txt -c n=8</pre>
<p>Output of clingo</p>	<pre>n=3 bishop(1,1) bishop(3,1) bishop(1,2) bishop(3,2) n=4 bishop(1,1) bishop(2,1) bishop(3,1) bishop(4,1) bishop(2,4) bishop(3,4) n=5 bishop(1,1) bishop(2,1) bishop(5,1) bishop(5,2) bishop(1,3) bishop(5,3) bishop(1,4) bishop(4,5) n=6 bishop(3,1) bishop(4,1) bishop(1,2) bishop(6,2) bishop(1,5) bishop(6,5) bishop(1,6) bishop(3,6) bishop(4,6) bishop(6,6) n=7 bishop(4,1) bishop(5,1) bishop(7,1) bishop(1,2) bishop(7,2) bishop(1,3) bishop(7,5) bishop(1,6) bishop(7,6) bishop(3,7) bishop(4,7) bishop(7,7) n=8 bishop(1,1) bishop(2,1) bishop(6,1) bishop(8,2) bishop(1,3) bishop(1,4) bishop(8,4) bishop(1,5) bishop(8,5) bishop(8,6) bishop(1,7) bishop(1,8) bishop(3,8) bishop(7,8)</pre>

Answer to Questions	<p>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.</p> <table border="1" data-bbox="443 348 1417 632"> <thead> <tr> <th>Value n</th><th>f(n)</th></tr> </thead> <tbody> <tr><td>3</td><td>4</td></tr> <tr><td>4</td><td>6</td></tr> <tr><td>5</td><td>8</td></tr> <tr><td>6</td><td>10</td></tr> <tr><td>7</td><td>12</td></tr> <tr><td>8</td><td>14</td></tr> </tbody> </table> <p>$f(n) = 2n - 2$</p>	Value n	f(n)	3	4	4	6	5	8	6	10	7	12	8	14
Value n	f(n)														
3	4														
4	6														
5	8														
6	10														
7	12														
8	14														

Problem 9

Input Program	<pre>%Partition all the n numbers into k sets {schur(N,1..k)}=1 :- N=1..n. %Make the subsets as sum-free :- schur(N,S), schur(M,S), schur(M+N,S).</pre>
Command Line	<pre>clingo schur.txt -c k=1 -c n=1 clingo schur.txt -c k=2 -c n=4 clingo schur.txt -c k=3 -c n=13 clingo schur.txt -c k=4 -c n=44</pre>
Output of clingo	<pre>K=1,N=1 Answer: 1 schur(1,1) SATISFIABLE Models : 1 Calls : 1 Time : 0.003s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s) CPU Time : 0.000s K=2, N=4 Answer: 1 schur(1,1) schur(2,2) schur(3,2) schur(4,1) SATISFIABLE Models : 1+ Calls : 1 Time : 0.007s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s) CPU Time : 0.000s K=3, N=13 Answer: 1 schur(1,1) schur(2,2) schur(3,2) schur(4,1) schur(5,3) schur(6,3) schur(7,1) schur(8,3) schur(9,3) schur(10,1) schur(11,2) schur(12,2) schur(13,1) SATISFIABLE Models : 1+ Calls : 1 Time : 0.010s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s) CPU Time : 0.000s</pre>

	<p>K=4, N=44</p> <p>Answer: 1</p> <p>schur(1,1) schur(2,3) schur(3,3) schur(4,1) schur(5,4) schur(6,4) schur(7,4) schur(8,4) schur(9,1) schur(10,3) schur(11,3) schur(12,1) schur(13,2) schur(14,2) schur(15,2) schur(16,3) schur(17,4) schur(18,4) schur(19,1) schur(20,2) schur(21,2) schur(22,2) schur(23,2) schur(24,2) schur(25,2) schur(26,1) schur(27,4) schur(28,4) schur(29,3) schur(30,3) schur(31,2) schur(32,2) schur(33,1) schur(34,3) schur(35,3) schur(36,1) schur(37,4) schur(38,4) schur(39,1) schur(40,4) schur(41,4) schur(42,3) schur(43,3) schur(44,1)</p> <p>SATISFIABLE</p> <p>Models : 1+</p> <p>Calls : 1</p> <p>Time : 0.294s (Solving: 0.27s 1st Model: 0.26s Unsat: 0.00s)</p> <p>CPU Time : 0.281s</p>								
Answer to Questions	<table border="1"> <tr> <td>Exact value of A(1)</td><td>N=1</td></tr> <tr> <td>Exact value of A(2)</td><td>N=4</td></tr> <tr> <td>Exact value of A(3)</td><td>N=13</td></tr> <tr> <td> 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. </td><td>N=44</td></tr> </table>	Exact value of A(1)	N=1	Exact value of A(2)	N=4	Exact value of A(3)	N=13	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.	N=44
Exact value of A(1)	N=1								
Exact value of A(2)	N=4								
Exact value of A(3)	N=13								
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.	N=44								