CSE 579
Programming Assignment 1
Solutions for Clingo Problems

Problem 1: Use clingo to find all solutions to the 8 queens' problem that have no queens in the 4x4=16 squares in the middle of the board.

Input Program	{pos(A,1..8)}=1 :- A=1..8. :- pos(A1, B), pos(A2, B), A1!=A2. :- pos(A1, B1), pos(A2, B2), A1!=A2, |A1-A2| = |B1-B2|. :- pos(A,B), A=3..6, B=3..6.
Command Line	clingo pa1_1.txt 0
Output of clingo	clingo version 5.4.0 Reading from pa1_1.txt Solving... Answer: 1 pos(5,7) pos(1,4) pos(2,6) pos(4,2) pos(3,8) pos(6,1) pos(7,3) pos(8,5) Answer: 2 pos(2,3) pos(3,1) pos(6,8) pos(4,7) pos(1,5) pos(5,2) pos(7,6) pos(8,4) Answer: 3 pos(2,4) pos(4,1) pos(5,8) pos(3,7) pos(1,6) pos(6,2) pos(7,5) pos(8,3) Answer: 4 pos(6,7) pos(1,3) pos(2,5) pos(3,2) pos(4,8) pos(5,1) pos(8,6) pos(7,4) SATISFIABLE Models: 4 Calls: 1 Time: 0.009s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s) CPU Time: 0.000s

Problem 2: Use clingo to find all solutions to the n-queens problem for n=3,4,5,6,7,8,9,10,11,12. Draw a table that lists the number of solutions and the times to compute all solutions. Use CPU time that clingo returns. Since the output is large, do not copy them into the submission.

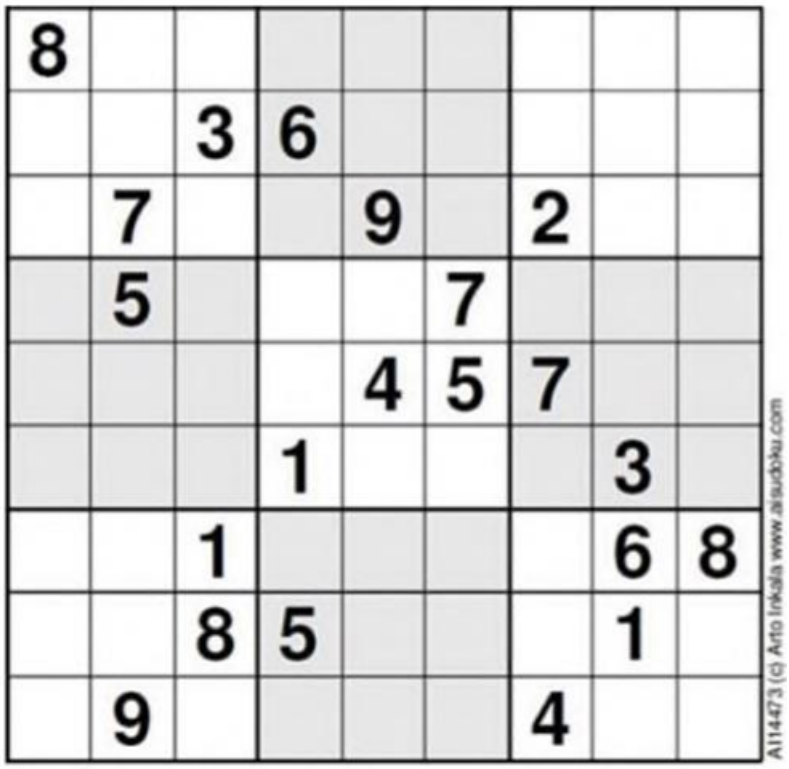
<table>
<tr><td>Input
Program</td><td>{pos(A,1..n)}=1 :- A=1..n.

:- pos(A1, B), pos(A2, B), A1!=A2.
:- pos(A1, B1), pos(A2, B2), A1!=A2, |A1-A2| = |B1-B2|.</td></tr>
<tr><td>Command
Line</td><td>You should write multiple command lines below.

clingo pa1_2.txt -c n=3 0
clingo pa1_2.txt -c n=4 0
clingo pa1_2.txt -c n=5 0
clingo pa1_2.txt -c n=6 0
clingo pa1_2.txt -c n=7 0
clingo pa1_2.txt -c n=8 0
clingo pa1_2.txt -c n=9 0
clingo pa1_2.txt -c n=10 0
clingo pa1_2.txt -c n=11 0
clingo pa1_2.txt -c n=12 0</td></tr>
<tr><td>Output
of clingo</td><td>Since the output is large, do not copy them into the submission.</td></tr>
<tr><td>Answer
to Questions</td><td>Draw a table that lists the number of solutions and the times to compute all solutions. Use CPU time that clingo returns.

<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>0.007s</td></tr>
<tr><td>5</td><td>10</td><td>0.014s</td></tr>
<tr><td>6</td><td>4</td><td>0.011s</td></tr>
<tr><td>7</td><td>40</td><td>0.037s</td></tr>
<tr><td>8</td><td>92</td><td>0.070s</td></tr>
<tr><td>9</td><td>352</td><td>0.280s</td></tr>
<tr><td>10</td><td>724</td><td>0.803s</td></tr>
<tr><td>11</td><td>2680</td><td>4.048s</td></tr>
<tr><td>12</td><td>14200</td><td>48.437s</td></tr>
</table>
</td></tr>
</table>

Problem 3: Use clingo to find all solutions to the so-called world's hardest sudoku problem below.



Input Program	1 { placeDigit(R,C,D) : R = 1..9, C=1..9, RBase <= R, R <= RBase+2, CBase <= C, C <= CBase+2 } 1 :- D = 1..9, RBase = 3*(0..2)+1, CBase = 3*(0..2)+1. %setting the constraints :- placeDigit(R,C,D1), placeDigit(R,C,D2), D1 != D2. :- placeDigit(R,C1,D), placeDigit(R,C2,D), C1 != C2. :- placeDigit(R1,C,D), placeDigit(R2,C,D), R1 != R2. placeDigit(1, 1, 8). placeDigit(3, 2, 7). placeDigit(4, 2, 5). placeDigit(9, 2, 9). placeDigit(2, 3, 3). placeDigit(7, 3, 1). placeDigit(8, 3, 8). placeDigit(2, 4, 6). placeDigit(6, 4, 1). placeDigit(8, 4, 5).

	placeDigit(3, 5, 9). placeDigit(5, 5, 4). placeDigit(4, 6, 7). placeDigit(5, 6, 5). placeDigit(3, 7, 2). placeDigit(5, 7, 7). placeDigit(9, 7, 4). placeDigit(6, 8, 3). placeDigit(7, 8, 6). placeDigit(8, 8, 1). placeDigit(7, 9, 8). %placed all the numbers as per the question
Command Line	clingo pa1_3.txt 0
Output of clingo	clingo version 5.4.0 Reading from pa1_3.txt Solving... Answer: 1 placeDigit(1,1,8) placeDigit(3,2,7) placeDigit(4,2,5) placeDigit(9,2,9) placeDigit(2,3,3) placeDigit(7,3,1) placeDigit(8,3,8) placeDigit(2,4,6) placeDigit(6,4,1) placeDigit(8,4,5) placeDigit(3,5,9) placeDigit(5,5,4) placeDigit(4,6,7) placeDigit(5,6,5) placeDigit(3,7,2) placeDigit(5,7,7) placeDigit(9,7,4) placeDigit(6,8,3) placeDigit(7,8,6) placeDigit(8,8,1) placeDigit(7,9,8) placeDigit(4,1,1) placeDigit(1,2,1) placeDigit(6,1,2) placeDigit(7,2,2) placeDigit(1,3,2) placeDigit(5,1,3) placeDigit(8,2,3) placeDigit(8,1,4) placeDigit(2,2,4) placeDigit(4,3,4) placeDigit(7,1,5) placeDigit(3,3,5) placeDigit(3,1,6) placeDigit(5,2,6) placeDigit(9,3,6) placeDigit(9,1,7) placeDigit(6,3,7) placeDigit(6,2,8) placeDigit(2,1,9) placeDigit(5,3,9) placeDigit(9,5,1) placeDigit(3,6,1) placeDigit(4,4,2) placeDigit(8,5,2) placeDigit(2,6,2) placeDigit(9,4,3) placeDigit(4,5,3) placeDigit(1,6,3) placeDigit(3,4,4) placeDigit(7,6,4) placeDigit(1,5,5) placeDigit(6,5,6) placeDigit(8,6,6) placeDigit(1,4,7) placeDigit(7,5,7) placeDigit(5,4,8) placeDigit(2,5,8) placeDigit(9,6,8) placeDigit(7,4,9) placeDigit(6,6,9) placeDigit(2,7,1) placeDigit(5,9,1) placeDigit(5,8,2) placeDigit(9,9,2) placeDigit(7,7,3) placeDigit(3,9,3) placeDigit(1,8,4) placeDigit(6,9,4) placeDigit(6,7,5) placeDigit(9,8,5) placeDigit(2,9,5) placeDigit(1,7,6) placeDigit(4,9,6) placeDigit(2,8,7) placeDigit(8,9,7) placeDigit(4,7,8) placeDigit(3,8,8) placeDigit(8,7,9) placeDigit(4,8,9) placeDigit(1,9,9) SATISFIABLE Models : 1 Calls : 1

	Time : 0.030s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.01s) CPU Time : 0.016s

Problem 4: Use clingo to find all solutions to the 16x16 Sudoku problem below.

9	14				3		5	15		2				7	1
6	12				14					10				5	11
4			7	6			13	16			1	2			9
	15	16		9	7					11	6		3	14	
	7	15											2	16	
5		13		14		15			10		3		1		8
	8		10		9	4	11	13	6	15		14		3	
16				5		3			14		9				6
15				16		10			9		13				14
	9		6		5	13	3	1	15	4		7		12	
2		8		15		14			16		12		5		13
	13	12											9	11	
	5	3		2	16					13	10		12	9	
8			4	12			1	6			7	15			3
10	1				15					16				6	2
11	2				8		14	3		1				10	7

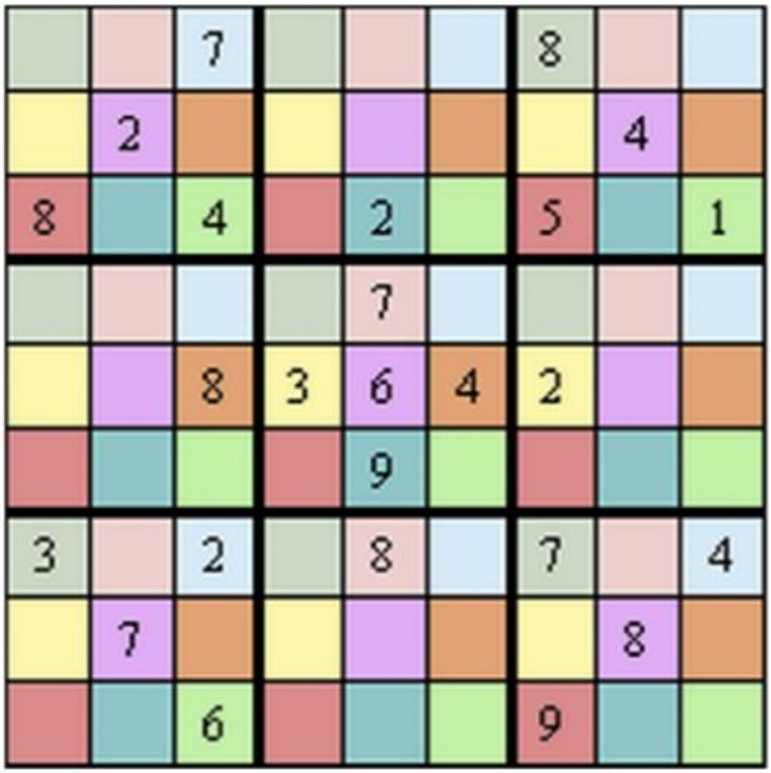
Input Program	1 { setEntry(R,C,Val) : R = 1..16, C = 1..16, R_start <= R, R <= R_start + 3, C_start <= C, C <= C_start + 3 } 1 :- Val = 1..16, R_start = 4*(0..3)+1, C_start = 4*(0..3)+1. %setting the constraints :- setEntry(R,C,Val1), setEntry(R,C,Val2), Val1 != Val2. :- setEntry(R,C1,Val), setEntry(R,C2,Val), C1 != C2. :- setEntry(R1,C,Val), setEntry(R2,C,Val), R1 != R2. setEntry(1, 1, 9). setEntry(1, 2, 14). setEntry(1, 6, 3). setEntry(1, 8, 5). setEntry(1, 9, 15). setEntry(1, 11, 2). setEntry(1, 15, 7). setEntry(1, 16, 1). setEntry(2, 1, 6). setEntry(2, 2, 12). setEntry(2, 6, 14). setEntry(2, 11, 10). setEntry(2, 15, 5). setEntry(2, 16, 11).

	setEntry(3, 1, 4). setEntry(3, 4, 7). setEntry(3, 5, 6). setEntry(3, 8, 13). setEntry(3, 9, 16). setEntry(3, 12, 1). setEntry(3, 13, 2). setEntry(3, 16, 9). setEntry(4, 2, 15). setEntry(4, 3, 16). setEntry(4, 5, 9). setEntry(4, 6, 7). setEntry(4, 11, 11). setEntry(4, 12, 6). setEntry(4, 14, 3). setEntry(4, 15, 14). setEntry(5, 2, 7). setEntry(5, 3, 15). setEntry(5, 14, 2). setEntry(5, 15, 16). setEntry(6, 1, 5). setEntry(6, 3, 13). setEntry(6, 5, 14). setEntry(6, 7, 15). setEntry(6, 10, 10). setEntry(6, 12, 3). setEntry(6, 14, 1). setEntry(6, 16, 8). setEntry(7, 2, 8). setEntry(7, 4, 10). setEntry(7, 6, 9). setEntry(7, 7, 4). setEntry(7, 8, 11). setEntry(7, 9, 13). setEntry(7, 10, 6). setEntry(7, 11, 15). setEntry(7, 13, 14). setEntry(7, 15, 3). setEntry(8, 1, 16). setEntry(8, 5, 5). setEntry(8, 7, 3). setEntry(8, 10, 14). setEntry(8, 12, 9). setEntry(8, 16, 6). setEntry(9, 1, 15). setEntry(9, 5, 16). setEntry(9, 7, 10). setEntry(9, 10, 9). setEntry(9, 12, 13). setEntry(9, 16, 14). setEntry(10, 2, 9). setEntry(10, 4, 6). setEntry(10, 6, 5). setEntry(10, 7, 13). setEntry(10, 8, 3). setEntry(10, 9, 1). setEntry(10, 10, 15). setEntry(10, 11, 4). setEntry(10, 13, 7). setEntry(10, 15, 12). setEntry(11, 1, 2). setEntry(11, 3, 8). setEntry(11, 5, 15). setEntry(11, 7, 14). setEntry(11, 10, 16). setEntry(11, 12, 12). setEntry(11, 14, 5). setEntry(11, 16, 13). setEntry(12, 2, 13). setEntry(12, 3, 12). setEntry(12, 14, 9). setEntry(12, 15, 11). setEntry(13, 2, 5). setEntry(13, 3, 3). setEntry(13, 5, 2). setEntry(13, 6, 16). setEntry(13, 11, 13). setEntry(13, 12, 10). setEntry(13, 14, 12). setEntry(13, 15, 9). setEntry(14, 1, 8). setEntry(14, 4, 4). setEntry(14, 5, 12). setEntry(14, 8, 1). setEntry(14, 9, 6). setEntry(14, 12, 7). setEntry(14, 13, 15). setEntry(14, 16, 3). setEntry(15, 1, 10). setEntry(15, 2, 1). setEntry(15, 6, 15). setEntry(15, 11, 16). setEntry(15, 15, 6). setEntry(15, 16, 2). setEntry(16, 1, 11). setEntry(16, 2, 2). setEntry(16, 6, 8). setEntry(16, 8, 14). setEntry(16, 9, 3). setEntry(16, 11, 1). setEntry(16, 15, 10). setEntry(16, 16, 7). %End
Command Line	clingo pa1_4.txt 0
Output of clingo	clingo version 5.4.0 Reading from pa1_4.txt Solving... Answer: 1 setEntry(1,1,9) setEntry(1,2,14) setEntry(1,6,3) setEntry(1,8,5) setEntry(1,9,15) setEntry(1,11,2) setEntry(1,15,7) setEntry(1,16,1) setEntry(2,1,6) setEntry(2,2,12) setEntry(2,6,14) setEntry(2,11,10) setEntry(2,15,5) setEntry(2,16,11) setEntry(3,1,4) setEntry(3,4,7) setEntry(3,5,6) setEntry(3,8,13) setEntry(3,9,16) setEntry(3,12,1)

	setEntry(3,13,2) setEntry(3,16,9) setEntry(4,2,15) setEntry(4,3,16) setEntry(4,5,9) setEntry(4,6,7) setEntry(4,11,11) setEntry(4,12,6) setEntry(4,14,3) setEntry(4,15,14) setEntry(5,2,7) setEntry(5,3,15) setEntry(5,14,2) setEntry(5,15,16) setEntry(6,1,5) setEntry(6,3,13) setEntry(6,5,14) setEntry(6,7,15) setEntry(6,10,10) setEntry(6,12,3) setEntry(6,14,1) setEntry(6,16,8) setEntry(7,2,8) setEntry(7,4,10) setEntry(7,6,9) setEntry(7,7,4) setEntry(7,8,11) setEntry(7,9,13) setEntry(7,10,6) setEntry(7,11,15) setEntry(7,13,14) setEntry(7,15,3) setEntry(8,1,16) setEntry(8,5,5) setEntry(8,7,3) setEntry(8,10,14) setEntry(8,12,9) setEntry(8,16,6) setEntry(9,1,15) setEntry(9,5,16) setEntry(9,7,10) setEntry(9,10,9) setEntry(9,12,13) setEntry(9,16,14) setEntry(10,2,9) setEntry(10,4,6) setEntry(10,6,5) setEntry(10,7,13) setEntry(10,8,3) setEntry(10,9,1) setEntry(10,10,15) setEntry(10,11,4) setEntry(10,13,7) setEntry(10,15,12) setEntry(11,1,2) setEntry(11,3,8) setEntry(11,5,15) setEntry(11,7,14) setEntry(11,10,16) setEntry(11,12,12) setEntry(11,14,5) setEntry(11,16,13) setEntry(12,2,13) setEntry(12,3,12) setEntry(12,14,9) setEntry(12,15,11) setEntry(13,2,5) setEntry(13,3,3) setEntry(13,5,2) setEntry(13,6,16) setEntry(13,11,13) setEntry(13,12,10) setEntry(13,14,12) setEntry(13,15,9) setEntry(14,1,8) setEntry(14,4,4) setEntry(14,5,12) setEntry(14,8,1) setEntry(14,9,6) setEntry(14,12,7) setEntry(14,13,15) setEntry(14,16,3) setEntry(15,1,10) setEntry(15,2,1) setEntry(15,6,15) setEntry(15,11,16) setEntry(15,15,6) setEntry(15,16,2) setEntry(16,1,11) setEntry(16,2,2) setEntry(16,6,8) setEntry(16,8,14) setEntry(16,9,3) setEntry(16,11,1) setEntry(16,15,10) setEntry(16,16,7) setEntry(12,1,1) setEntry(2,3,1) setEntry(8,4,1) setEntry(7,3,2) setEntry(4,4,2) setEntry(5,1,3) setEntry(9,2,3) setEntry(2,4,3) setEntry(11,2,4) setEntry(8,3,4) setEntry(3,3,5) setEntry(9,4,5) setEntry(6,2,6) setEntry(16,3,6) setEntry(13,1,7) setEntry(9,3,7) setEntry(1,4,8) setEntry(15,3,9) setEntry(6,4,9) setEntry(3,2,10) setEntry(10,3,10) setEntry(8,2,11) setEntry(1,3,11) setEntry(11,4,11) setEntry(7,1,12) setEntry(16,4,12) setEntry(4,1,13) setEntry(15,4,13) setEntry(10,1,14) setEntry(14,3,14) setEntry(5,4,14) setEntry(13,4,15) setEntry(14,2,16) setEntry(12,4,16) setEntry(7,5,1) setEntry(9,6,1) setEntry(4,7,1) setEntry(8,6,2) setEntry(2,7,2) setEntry(12,8,2) setEntry(15,5,3) setEntry(2,5,4) setEntry(12,6,4) setEntry(15,8,4) setEntry(14,7,5) setEntry(11,6,6) setEntry(5,7,6) setEntry(13,8,6) setEntry(12,5,7) setEntry(15,7,7) setEntry(8,8,7) setEntry(5,5,8) setEntry(12,7,8) setEntry(4,8,8) setEntry(16,7,9) setEntry(11,8,9) setEntry(1,5,10) setEntry(14,6,10) setEntry(5,8,10) setEntry(10,5,11) setEntry(3,6,11) setEntry(13,7,11) setEntry(6,6,12) setEntry(3,7,12) setEntry(9,8,12) setEntry(16,5,13) setEntry(5,6,13) setEntry(2,8,15) setEntry(1,7,16) setEntry(6,8,16) setEntry(5,10,1) setEntry(6,9,2) setEntry(14,10,2) setEntry(10,12,2) setEntry(3,10,3) setEntry(11,11,3) setEntry(5,9,4) setEntry(16,10,4) setEntry(1,12,4) setEntry(4,9,5)

	setEntry(12,10,5) setEntry(5,11,5) setEntry(15,12,5) setEntry(12,11,6) setEntry(11,9,7) setEntry(2,10,7) setEntry(6,11,7) setEntry(8,9,8) setEntry(13,10,8) setEntry(9,11,8) setEntry(2,12,8) setEntry(2,9,9) setEntry(14,11,9) setEntry(12,9,10) setEntry(9,9,11) setEntry(15,10,11) setEntry(5,12,11) setEntry(15,9,12) setEntry(4,10,12) setEntry(8,11,12) setEntry(1,10,13) setEntry(13,9,14) setEntry(3,11,14) setEntry(12,12,14) setEntry(16,12,15) setEntry(7,12,16) setEntry(13,13,1) setEntry(11,15,1) setEntry(9,15,2) setEntry(12,13,3) setEntry(4,13,4) setEntry(9,14,4) setEntry(6,15,4) setEntry(13,16,4) setEntry(16,13,5) setEntry(7,16,5) setEntry(9,13,6) setEntry(1,14,6) setEntry(7,14,7) setEntry(15,13,8) setEntry(10,14,8) setEntry(3,15,8) setEntry(5,13,9) setEntry(11,13,10) setEntry(8,14,10) setEntry(4,16,10) setEntry(6,13,11) setEntry(14,14,11) setEntry(1,13,12) setEntry(5,16,12) setEntry(8,13,13) setEntry(2,14,13) setEntry(14,15,13) setEntry(15,14,14) setEntry(3,14,15) setEntry(8,15,15) setEntry(12,16,15) setEntry(2,13,16) setEntry(16,14,16) setEntry(10,16,16) SATISFIABLE Models : 1 Calls : 1 Time : 0.109s (Solving: 0.02s 1st Model: 0.00s Unsat: 0.02s) CPU Time : 0.078s

Problem 5: Use clingo to find all solutions to the Offset Sudoku problem below.

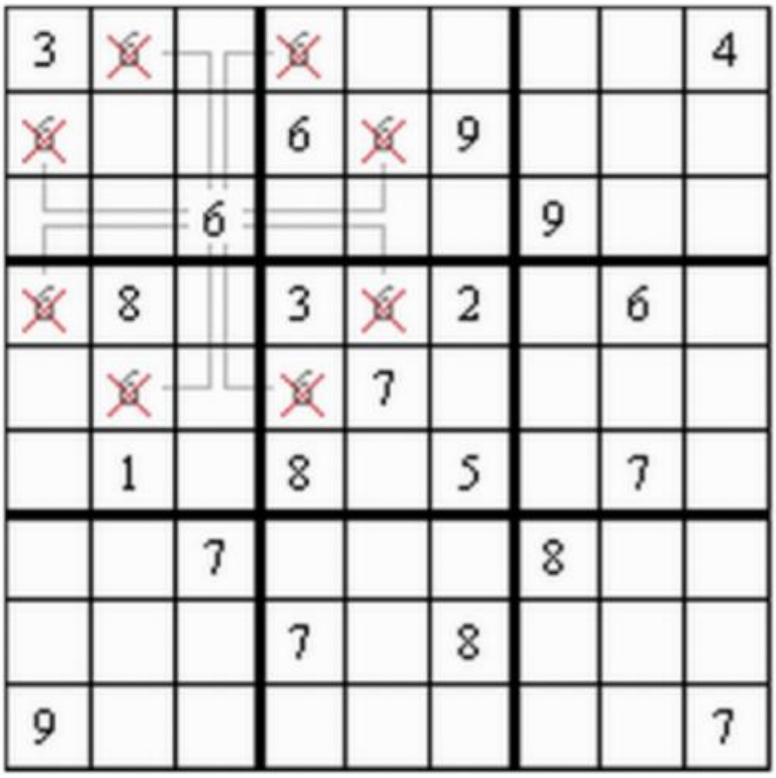


Input Program	% Ensure every number from 1 to 9 appears in each 3x3 grid 1 { placeNumber(R,C,Num) : R = 1..9, C = 1..9, R_start <= R, R <= R_start + 2, C_start <= C, C <= C_start + 2 } 1 :- Num = 1..9, R_start = 3*(0..2)+1, C_start = 3*(0..2)+1. % Prohibit duplicate numbers in the same row or column :- placeNumber(R,C,N1), placeNumber(R,C,N2), N1 != N2. % Ensure no duplicate numbers in any column :- placeNumber(R1,C,N), placeNumber(R2,C,N), R1 != R2. % Ensure no duplicate numbers in any row :- placeNumber(R,C1,N), placeNumber(R,C2,N), C1 != C2. % Enforce unique numbers within each 3x3 grid beyond the first rule :- placeNumber(R1,C1,N), placeNumber(R2,C2,N), R1 \ 3 == R2 \ 3, C1 \ 3 == C2 \ 3, 1 { R1 != R2; C1 != C2 }. % Pre-filled numbers placeNumber(1, 3, 7). placeNumber(1, 7, 8). placeNumber(2, 2, 2). placeNumber(2, 8, 4). placeNumber(3, 1, 8). placeNumber(3, 3, 4). placeNumber(3, 5, 2).

	placeNumber(3, 7, 5). placeNumber(3, 9, 1). placeNumber(4, 5, 7). placeNumber(5, 3, 8). placeNumber(5, 4, 3). placeNumber(5, 5, 6). placeNumber(5, 6, 4). placeNumber(5, 7, 2). placeNumber(6, 5, 9). placeNumber(7, 1, 3). placeNumber(7, 3, 2). placeNumber(7, 5, 8). placeNumber(7, 7, 7). placeNumber(7, 9, 4). placeNumber(8, 2, 7). placeNumber(8, 8, 8). placeNumber(9, 3, 6). placeNumber(9, 7, 9).
Command Line	clingo pa1_5.txt 0
Output of clingo	clingo version 5.4.0 Reading from pa1_5.txt Solving... Answer: 1 placeNumber(1,3,7) placeNumber(1,7,8) placeNumber(2,2,2) placeNumber(2,8,4) placeNumber(3,1,8) placeNumber(3,3,4) placeNumber(3,5,2) placeNumber(3,7,5) placeNumber(3,9,1) placeNumber(4,5,7) placeNumber(5,3,8) placeNumber(5,4,3) placeNumber(5,5,6) placeNumber(5,6,4) placeNumber(5,7,2) placeNumber(6,5,9) placeNumber(7,1,3) placeNumber(7,3,2) placeNumber(7,5,8) placeNumber(7,7,7) placeNumber(7,9,4) placeNumber(8,2,7) placeNumber(8,8,8) placeNumber(9,3,6) placeNumber(9,7,9) placeNumber(4,3,1) placeNumber(4,6,8) placeNumber(4,9,6) placeNumber(7,6,5) placeNumber(4,1,2) placeNumber(4,4,5) placeNumber(4,7,4) placeNumber(7,4,9) placeNumber(5,2,9) placeNumber(5,8,1) placeNumber(8,5,3) placeNumber(6,1,6) placeNumber(6,4,1) placeNumber(6,7,3) placeNumber(9,1,4) placeNumber(9,4,2) placeNumber(6,3,5) placeNumber(6,6,2) placeNumber(6,9,8) placeNumber(9,6,7) placeNumber(9,9,3) placeNumber(6,2,4) placeNumber(6,8,7) placeNumber(9,2,8) placeNumber(9,5,1) placeNumber(9,8,5) placeNumber(1,2,5) placeNumber(1,5,4) placeNumber(1,8,2) placeNumber(7,2,1) placeNumber(7,8,6) placeNumber(2,3,3) placeNumber(2,6,1) placeNumber(2,9,7) placeNumber(8,3,9) placeNumber(8,6,6) placeNumber(8,9,2) placeNumber(2,1,9) placeNumber(2,4,8) placeNumber(2,7,6) placeNumber(8,1,5) placeNumber(8,4,4) placeNumber(8,7,1) placeNumber(2,5,5) placeNumber(3,2,6) placeNumber(3,8,3) placeNumber(1,1,1) placeNumber(1,4,6) placeNumber(1,6,3) placeNumber(1,9,9) placeNumber(4,2,3) placeNumber(4,8,9) placeNumber(3,6,9) placeNumber(3,4,7) placeNumber(5,1,7) placeNumber(5,9,5) SATISFIABLE Models: 1

	Calls: 1 Time: 0.040s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.01s) CPU Time: 0.031s

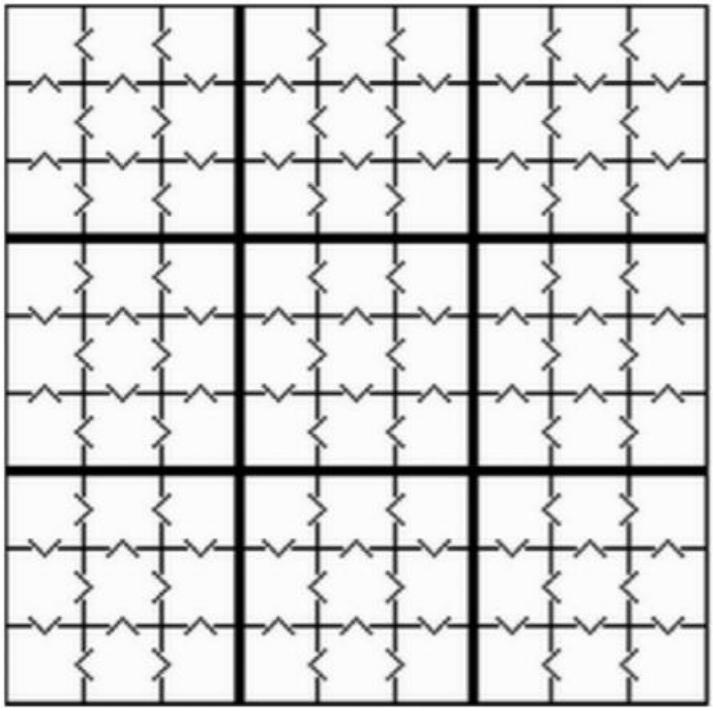
Problem 6: Use clingo to find all solutions to the Anti-Knight Sudoku problem presented below.



Input Program	% Every number from 1 to 9 must appear in each 3x3 subgrid 1 { puzzle(R,C,Num) :R=1..9,C=1..9,RStart<=R,R<=RStart+2,CStart<=C,C<=CStart+2 } 1 :- Num=1..9, RStart = 3*(0..2)+1, CStart = 3*(0..2)+1. % Prevent any two cells in the same row or column from having the same number :- puzzle(Row,Col,Num1), puzzle(Row,Col,Num2), Num1!=Num2. % Ensure no duplicate numbers in any column :- puzzle(Row1,Col,Num), puzzle(Row2,Col,Num), Row1!=Row2. % Ensure no duplicate numbers in any row :- puzzle(Row,Col1,Num), puzzle(Row,Col2,Num), Col1!=Col2. % Knight's move constraint: a number cannot appear in a cell that is a knight's move away from a cell with the same number :- puzzle(Row,Col,Num), puzzle(Row-2,Col-1,Num). :- puzzle(Row,Col,Num), puzzle(Row-2,Col+1,Num). :- puzzle(Row,Col,Num), puzzle(Row-1,Col-2,Num).

	:- puzzle(Row,Col,Num), puzzle(Row-1,Col+2,Num). :- puzzle(Row,Col,Num), puzzle(Row+1,Col-2,Num). :- puzzle(Row,Col,Num), puzzle(Row+1,Col+2,Num). :- puzzle(Row,Col,Num), puzzle(Row+2,Col-1,Num). :- puzzle(Row,Col,Num), puzzle(Row+2,Col+1,Num). % Initial clues given for the puzzle puzzle(1, 1, 3). puzzle(1, 9, 4). puzzle(2, 4, 6). puzzle(2, 6, 9). puzzle(3, 3, 6). puzzle(3, 7, 9). puzzle(4, 2, 8). puzzle(4, 4, 3). puzzle(4, 6, 2). puzzle(4, 8, 6). puzzle(5, 5, 7). puzzle(6, 2, 1). puzzle(6, 4, 8). puzzle(6, 6, 5). puzzle(6, 8, 7). puzzle(7, 3, 7). puzzle(7, 7, 8). puzzle(8, 4, 7). puzzle(8, 6, 8). puzzle(9, 1, 9). puzzle(9, 9, 7).
Command Line	clingo pa1_6.txt 0
Output of clingo	clingo version 5.4.0 Reading from pa1_6.txt Solving... Answer: 1 puzzle(1,1,3) puzzle(1,9,4) puzzle(2,4,6) puzzle(2,6,9) puzzle(3,3,6) puzzle(3,7,9) puzzle(4,2,8) puzzle(4,4,3) puzzle(4,6,2) puzzle(4,8,6) puzzle(5,5,7) puzzle(6,2,1) puzzle(6,4,8) puzzle(6,6,5) puzzle(6,8,7) puzzle(7,3,7) puzzle(7,7,8) puzzle(8,4,7) puzzle(8,6,8) puzzle(9,1,9) puzzle(9,9,7) puzzle(3,2,7) puzzle(1,2,9) puzzle(1,3,1) puzzle(3,6,1) puzzle(1,5,2) puzzle(3,5,3) puzzle(3,4,4) puzzle(1,4,5) puzzle(3,9,2) puzzle(3,8,5) puzzle(1,7,6) puzzle(1,6,7) puzzle(1,8,8) puzzle(2,2,2) puzzle(5,3,3) puzzle(2,1,4) puzzle(6,3,4) puzzle(5,2,5) puzzle(4,1,7) puzzle(3,1,8) puzzle(4,3,9) puzzle(4,5,1) puzzle(5,6,4) puzzle(2,3,5) puzzle(6,5,6) puzzle(2,5,8) puzzle(5,4,9) puzzle(5,8,1) puzzle(5,7,2) puzzle(2,8,3) puzzle(6,7,3) puzzle(4,7,4) puzzle(4,9,5) puzzle(2,7,7) puzzle(5,9,8) puzzle(6,9,9) puzzle(7,1,1) puzzle(6,1,2) puzzle(8,3,2) puzzle(9,2,3) puzzle(7,2,4) puzzle(5,1,6) puzzle(8,2,6) puzzle(9,3,8) puzzle(9,4,1) puzzle(7,4,2) puzzle(7,6,3) puzzle(9,5,4) puzzle(7,5,5) puzzle(9,6,6) puzzle(8,5,9) puzzle(8,7,1) puzzle(9,8,2) puzzle(8,9,3) puzzle(8,8,4) puzzle(9,7,5) puzzle(7,9,6) puzzle(7,8,9) puzzle(2,9,1) puzzle(8,1,5) SATISFIABLE Models : 1 Calls : 1 Time : 0.035s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.01s) CPU Time : 0.031s

Problem 7: Use clingo to find all solutions to the Greater-Than Sudoku problem presented below.



Input Program	%all numbers from 1 to 9 must be present in 3x3 grid 1 { cellValue(R,C,Num) :R=1..9,C=1..9,R1<=R,R<=R1+2,C1<=C,C<=C1+2 } 1 :- Num=1..9, R1 = 3*(0..2)+1, C1 = 3*(0..2)+1. % no two cells in the same R or C must have same number :- cellValue(R,C,Num1), cellValue(R,C,Num2), Num1!=Num2. % no two cells in the same C must have same number :- cellValue(R,C1,Num), cellValue(R,C2,Num), C1!=C2. % no two cells in the same R must have same number :- cellValue(R1,C,Num), cellValue(R2,C,Num), R1!=R2. %greater than :- cellValue(R1,C1,Num1), cellValue(R2,C2,Num2), isGreaterThan(R1,C1,R2,C2), Num1 <=Num2. isGreaterThan(1, 2, 1, 1). isGreaterThan(1, 3, 1, 2). isGreaterThan(1, 3, 2, 3). isGreaterThan(1, 4, 1, 5). isGreaterThan(1, 6, 1, 5). isGreaterThan(1, 6, 2, 6). isGreaterThan(1, 7, 2, 7). isGreaterThan(1, 8, 2, 8). isGreaterThan(1, 9, 1, 8). isGreaterThan(1, 9, 2, 9).

	isGreaterThan(2, 1, 1, 1). isGreaterThan(2, 2, 2, 1). isGreaterThan(2, 2, 2, 3). isGreaterThan(2, 2, 1, 2). isGreaterThan(2, 2, 3, 2). isGreaterThan(2, 3, 3, 3). isGreaterThan(2, 4, 3, 4). isGreaterThan(2, 5, 1, 5). isGreaterThan(2, 5, 2, 4). isGreaterThan(2, 5, 2, 6). isGreaterThan(2, 5, 3, 5). isGreaterThan(2, 6, 3, 6). isGreaterThan(2, 8, 2, 7). isGreaterThan(2, 9, 2, 8). isGreaterThan(2, 9, 3, 9). isGreaterThan(3, 1, 3, 2). isGreaterThan(3, 4, 3, 5). isGreaterThan(3, 5, 3, 6). isGreaterThan(3, 7, 2, 7). isGreaterThan(3, 7, 3, 8). isGreaterThan(3, 8, 2, 8). isGreaterThan(4, 1, 4, 2). isGreaterThan(4, 1, 5, 1). isGreaterThan(4, 3, 4, 2). isGreaterThan(4, 3, 5, 3). isGreaterThan(4, 6, 4, 5). isGreaterThan(4, 6, 5, 6). isGreaterThan(4, 7, 4, 8). isGreaterThan(4, 9, 4, 8). isGreaterThan(5, 2, 5, 1). isGreaterThan(5, 2, 5, 3). isGreaterThan(5, 2, 4, 2). isGreaterThan(5, 2, 6, 2). isGreaterThan(5, 4, 4, 4). isGreaterThan(5, 4, 6, 4). isGreaterThan(5, 4, 5, 5). isGreaterThan(5, 5, 4, 5). isGreaterThan(5, 5, 6, 5). isGreaterThan(5, 6, 5, 5). isGreaterThan(5, 7, 4, 7). isGreaterThan(5, 7, 5, 8). isGreaterThan(5, 8, 5, 9). isGreaterThan(5, 8, 4, 8). isGreaterThan(5, 9, 4, 9). isGreaterThan(6, 1, 5, 1). isGreaterThan(6, 2, 6, 1). isGreaterThan(6, 2, 6, 3). isGreaterThan(6, 5, 6, 4). isGreaterThan(6, 6, 6, 5). isGreaterThan(6, 6, 5, 6). isGreaterThan(6, 7, 5, 7). isGreaterThan(6, 8, 6, 7). isGreaterThan(6, 8, 6, 9). isGreaterThan(6, 8, 5, 8). isGreaterThan(6, 9, 5, 9). isGreaterThan(7, 1, 7, 2). isGreaterThan(7, 1, 8, 1). isGreaterThan(7, 3, 7, 2). isGreaterThan(7, 3, 8, 3). isGreaterThan(7, 4, 7, 5). isGreaterThan(7, 4, 8, 4). isGreaterThan(7, 6, 8, 6). isGreaterThan(7, 6, 7, 5). isGreaterThan(7, 7, 8, 7). isGreaterThan(7, 8, 7, 7). isGreaterThan(7, 8, 7, 9). isGreaterThan(8, 1, 9, 1). isGreaterThan(8, 1, 8, 2). isGreaterThan(8, 2, 8, 3). isGreaterThan(8, 2, 7, 2). isGreaterThan(8, 5, 8, 4). isGreaterThan(8, 5, 8, 6). isGreaterThan(8, 5, 7, 5). isGreaterThan(8, 6, 9, 6). isGreaterThan(8, 7, 9, 7). isGreaterThan(8, 8, 8, 7). isGreaterThan(8, 8, 7, 8). isGreaterThan(8, 8, 9, 8). isGreaterThan(8, 9, 8, 8). isGreaterThan(8, 9, 7, 9). isGreaterThan(8, 9, 9, 9). isGreaterThan(9, 2, 8, 2). isGreaterThan(9, 2, 9, 1). isGreaterThan(9, 2, 9, 3). isGreaterThan(9, 3, 8, 3). isGreaterThan(9, 4, 8, 4). isGreaterThan(9, 5, 9, 4). isGreaterThan(9, 5, 9, 6). isGreaterThan(9, 5, 8, 5). isGreaterThan(9, 8, 9, 7). isGreaterThan(9, 9, 9, 8).
Command Line	clingo pa1_7.txt 0
Output of clingo	clingo version 5.4.0 Reading from pa1_7.txt Solving... Answer: 1

	isGreaterThan(1,2,1,1) isGreaterThan(1,3,1,2) isGreaterThan(1,3,2,3) isGreaterThan(1,4,1,5) isGreaterThan(1,6,1,5) isGreaterThan(1,6,2,6) isGreaterThan(1,7,2,7) isGreaterThan(1,8,2,8) isGreaterThan(1,9,1,8) isGreaterThan(1,9,2,9) isGreaterThan(2,1,1,1) isGreaterThan(2,2,2,1) isGreaterThan(2,2,2,3) isGreaterThan(2,2,1,2) isGreaterThan(2,2,3,2) isGreaterThan(2,3,3,3) isGreaterThan(2,4,3,4) isGreaterThan(2,5,1,5) isGreaterThan(2,5,2,4) isGreaterThan(2,5,2,6) isGreaterThan(2,5,3,5) isGreaterThan(2,6,3,6) isGreaterThan(2,8,2,7) isGreaterThan(2,9,2,8) isGreaterThan(2,9,3,9) isGreaterThan(3,1,3,2) isGreaterThan(3,4,3,5) isGreaterThan(3,5,3,6) isGreaterThan(3,7,2,7) isGreaterThan(3,7,3,8) isGreaterThan(3,8,2,8) isGreaterThan(4,1,4,2) isGreaterThan(4,1,5,1) isGreaterThan(4,3,4,2) isGreaterThan(4,3,5,3) isGreaterThan(4,6,4,5) isGreaterThan(4,6,5,6) isGreaterThan(4,7,4,8) isGreaterThan(4,9,4,8) isGreaterThan(5,2,5,1) isGreaterThan(5,2,5,3) isGreaterThan(5,2,4,2) isGreaterThan(5,2,6,2) isGreaterThan(5,4,4,4) isGreaterThan(5,4,6,4) isGreaterThan(5,4,5,5) isGreaterThan(5,5,4,5) isGreaterThan(5,5,6,5) isGreaterThan(5,6,5,5) isGreaterThan(5,7,4,7) isGreaterThan(5,7,5,8) isGreaterThan(5,8,5,9) isGreaterThan(5,8,4,8) isGreaterThan(5,9,4,9) isGreaterThan(6,1,5,1) isGreaterThan(6,2,6,1) isGreaterThan(6,2,6,3) isGreaterThan(6,5,6,4) isGreaterThan(6,6,6,5) isGreaterThan(6,6,5,6) isGreaterThan(6,7,5,7) isGreaterThan(6,8,6,7) isGreaterThan(6,8,6,9) isGreaterThan(6,8,5,8) isGreaterThan(6,9,5,9) isGreaterThan(7,1,7,2) isGreaterThan(7,1,8,1) isGreaterThan(7,3,7,2) isGreaterThan(7,3,8,3) isGreaterThan(7,4,7,5) isGreaterThan(7,4,8,4) isGreaterThan(7,6,8,6) isGreaterThan(7,6,7,5) isGreaterThan(7,7,8,7) isGreaterThan(7,8,7,7) isGreaterThan(7,8,7,9) isGreaterThan(8,1,9,1) isGreaterThan(8,1,8,2) isGreaterThan(8,2,8,3) isGreaterThan(8,2,7,2) isGreaterThan(8,5,8,4) isGreaterThan(8,5,8,6) isGreaterThan(8,5,7,5) isGreaterThan(8,6,9,6) isGreaterThan(8,7,9,7) isGreaterThan(8,8,8,7) isGreaterThan(8,8,7,8) isGreaterThan(8,8,9,8) isGreaterThan(8,9,8,8) isGreaterThan(8,9,7,9) isGreaterThan(8,9,9,9) isGreaterThan(9,2,8,2) isGreaterThan(9,2,9,1) isGreaterThan(9,2,9,3) isGreaterThan(9,3,8,3) isGreaterThan(9,4,8,4) isGreaterThan(9,5,9,4) isGreaterThan(9,5,9,6) isGreaterThan(9,5,8,5) isGreaterThan(9,8,9,7) isGreaterThan(9,9,9,8) cellValue(1,1,2) cellValue(1,2,3) cellValue(1,3,9) cellValue(2,3,6) cellValue(1,5,1) cellValue(1,4,5) cellValue(1,6,4) cellValue(2,6,3) cellValue(2,7,1) cellValue(1,7,6) cellValue(2,8,2) cellValue(1,8,7) cellValue(1,9,8) cellValue(2,9,5) cellValue(2,1,4) cellValue(2,2,7) cellValue(3,2,1) cellValue(3,3,5) cellValue(3,4,7) cellValue(2,4,8) cellValue(2,5,9) cellValue(3,5,6) cellValue(3,6,2) cellValue(3,9,4) cellValue(3,1,8) cellValue(3,7,9) cellValue(3,8,3) cellValue(4,2,6) cellValue(4,1,9) cellValue(5,1,1) cellValue(4,3,7) cellValue(5,3,2) cellValue(4,5,4) cellValue(4,6,8) cellValue(5,6,6) cellValue(4,8,1) cellValue(4,7,5) cellValue(4,9,2) cellValue(5,2,8) cellValue(6,2,5) cellValue(4,4,3)

	cellValue(5,4,9) cellValue(6,4,1) cellValue(5,5,5) cellValue(6,5,2) cellValue(5,7,7) cellValue(5,8,4) cellValue(5,9,3) cellValue(6,1,3) cellValue(6,3,4) cellValue(6,6,7) cellValue(6,7,8) cellValue(6,8,9) cellValue(6,9,6) cellValue(7,2,2) cellValue(7,1,7) cellValue(8,1,6) cellValue(7,3,8) cellValue(8,3,1) cellValue(7,5,3) cellValue(7,4,6) cellValue(8,4,2) cellValue(8,6,5) cellValue(7,6,9) cellValue(8,7,3) cellValue(7,7,4) cellValue(7,8,5) cellValue(7,9,1) cellValue(9,1,5) cellValue(8,2,4) cellValue(8,5,7) cellValue(9,6,1) cellValue(9,7,2) cellValue(8,8,8) cellValue(9,8,6) cellValue(8,9,9) cellValue(9,9,7) cellValue(9,2,9) cellValue(9,3,3) cellValue(9,4,4) cellValue(9,5,8) SATISFIABLE Models : 1 Calls : 1 Time : 0.437s (Solving: 0.34s 1st Model: 0.28s Unsat: 0.07s) CPU Time : 0.297s

Problem 8: Use clingo to determine how many bishops can be placed on a chessboard so that they do not attack each other.
(a) Find the maximum value of bishops when the chessboard is n by n, where n is 3, 4, 5, 6, 7,8.
(b) Infer the general function f(n) that returns the maximum value of bishops.

Input Program	% every rank must have at least one bishop {bishop(Rank,1..n)} :- Rank=1..n. % no diagonal must have 2 bishops attacking each other :- bishop(Rank1,File1), bishop(Rank2,File2), Rank1 != Rank2, |Rank1-Rank2|==|File1-File2|. %op #maximize{ 1, Rank, File : bishop(Rank, File)}.
Command Line	You should write multiple command lines below. clingo pa1_8.txt -c n=3 clingo pa1_8.txt -c n=4 clingo pa1_8.txt -c n=5 clingo pa1_8.txt -c n=6 clingo pa1_8.txt -c n=7 clingo pa1_8.txt -c n=8
Output of clingo	**For n = 3 (3 ranks, 3 files)** clingo version 5.4.0 Reading from pa1_8.txt Solving... Answer: 1 Optimization: 0 Answer: 2 bishop(1,1) Optimization: -1 Answer: 3 bishop(1,1) bishop(3,1) Optimization: -2 Answer: 4 bishop(1,1) bishop(3,1) bishop(1,2) Optimization: -3 Answer: 5 bishop(1,1) bishop(3,1) bishop(1,2) bishop(3,2) Optimization: -4 OPTIMUM FOUND

	Models: 5 Optimum: yes Optimization: -4 Calls: 1 Time: 0.017s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s) CPU Time : 0.000s **For n = 4 (4 ranks, 4 files)** clingo version 5.4.0 Reading from pa1_8.txt Solving... Answer: 1 Optimization: 0 Answer: 2 bishop(3,4) Optimization: -1 Answer: 3 bishop(2,4) bishop(3,4) Optimization: -2 Answer: 4 bishop(4,1) bishop(2,4) bishop(3,4) Optimization: -3 Answer: 5 bishop(4,1) bishop(2,4) bishop(3,4) bishop(4,4) Optimization: -4 Answer: 6 bishop(3,1) bishop(4,1) bishop(2,4) bishop(3,4) bishop(4,4) Optimization: -5 Answer: 7 bishop(1,1) bishop(2,1) bishop(3,1) bishop(4,1) bishop(2,4) bishop(3,4) Optimization: -6 OPTIMUM FOUND Models: 7 Optimum: yes Optimization: -6 Calls: 1 Time: 0.010s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.01s) CPU Time: 0.000s **For n = 5 (5 ranks, 5 files)** clingo version 5.4.0 Reading from pa1_8.txt

	Solving... Answer: 1 Optimization: 0 Answer: 2 bishop(1,5) Optimization: -1 Answer: 3 bishop(2,1) bishop(1,5) Optimization: -2 Answer: 4 bishop(2,1) bishop(1,5) bishop(2,5) Optimization: -3 Answer: 5 bishop(2,1) bishop(5,1) bishop(2,5) bishop(4,5) Optimization: -4 Answer: 6 bishop(2,1) bishop(5,1) bishop(2,5) bishop(4,5) bishop(5,5) Optimization: -5 Answer: 7 bishop(2,1) bishop(1,5) bishop(2,5) bishop(3,5) bishop(4,5) bishop(5,5) Optimization: -6 Answer: 8 bishop(1,1) bishop(2,1) bishop(3,1) bishop(5,1) bishop(2,5) bishop(3,5) bishop(4,5) Optimization: -7 Answer: 9 bishop(1,1) bishop(2,1) bishop(5,1) bishop(5,2) bishop(1,3) bishop(5,3) bishop(1,4) bishop(4,5) Optimization: -8 OPTIMUM FOUND Models: 9 Optimum: yes Optimization: -8 Calls: 1 Time: 0.045s (Solving: 0.04s 1st Model: 0.00s Unsat: 0.01s) CPU Time: 0.000s **For n = 6 (6 ranks, 6 files)** clingo version 5.4.0 Reading from pa1_8.txt Solving... Answer: 1

	Optimization: 0 Answer: 2 bishop(6,1) Optimization: -1 Answer: 3 bishop(6,1) bishop(6,6) Optimization: -2 Answer: 4 bishop(6,1) bishop(3,6) bishop(6,6) Optimization: -3 Answer: 5 bishop(6,1) bishop(1,2) bishop(3,6) bishop(6,6) Optimization: -4 Answer: 6 bishop(6,1) bishop(1,2) bishop(1,5) bishop(3,6) bishop(6,6) Optimization: -5 Answer: 7 bishop(6,1) bishop(1,2) bishop(1,5) bishop(3,6) bishop(4,6) bishop(6,6) Optimization: -6 Answer: 8 bishop(1,2) bishop(6,5) bishop(1,6) bishop(2,6) bishop(3,6) bishop(4,6) bishop(6,6) Optimization: -7 Answer: 9 bishop(6,1) bishop(1,2) bishop(6,2) bishop(1,5) bishop(6,5) bishop(3,6) bishop(4,6) bishop(6,6) Optimization: -8 Answer: 10 bishop(6,1) bishop(1,2) bishop(6,2) bishop(1,3) bishop(6,4) bishop(1,5) bishop(6,5) bishop(3,6) bishop(6,6) Optimization: -9 Answer: 11 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) Optimization: -10 OPTIMUM FOUND Models : 11 Optimum : yes Optimization : -10 Calls : 1 Time : 0.105s (Solving: 0.10s 1st Model: 0.00s Unsat: 0.06s) CPU Time : 0.063s

	For n = 7 (7 ranks, 7 files) clingo version 5.4.0 Reading from pa1_8.txt Solving... Answer: 1 Optimization: 0 Answer: 2 bishop(1,3) Optimization: -1 Answer: 3 bishop(1,3) bishop(7,5) Optimization: -2 Answer: 4 bishop(7,1) bishop(1,3) bishop(7,5) Optimization: -3 Answer: 5 bishop(7,1) bishop(1,3) bishop(7,4) bishop(7,5) Optimization: -4 Answer: 6 bishop(7,1) bishop(1,3) bishop(7,4) bishop(7,5) bishop(3,7) Optimization: -5 Answer: 7 bishop(7,1) bishop(1,3) bishop(7,5) bishop(1,6) bishop(3,7) bishop(4,7) Optimization: -6 Answer: 8 bishop(7,1) bishop(1,3) bishop(7,5) bishop(1,6) bishop(3,7) bishop(4,7) bishop(7,7) Optimization: -7 Answer: 9 bishop(7,1) bishop(1,2) bishop(1,3) bishop(7,5) bishop(1,6) bishop(3,7) bishop(4,7) bishop(7,7) Optimization: -8 Answer: 10 bishop(7,1) bishop(1,2) bishop(1,3) bishop(7,5) bishop(1,6) bishop(7,6) bishop(3,7) bishop(4,7) bishop(7,7) Optimization: -9 Answer: 11 bishop(7,1) bishop(1,2) bishop(7,2) bishop(1,3) bishop(1,5) bishop(7,5) bishop(1,6) bishop(7,6) bishop(4,7) bishop(7,7) Optimization: -10 Answer: 12

	bishop(3,1) bishop(5,1) bishop(1,2) bishop(7,2) bishop(7,4) bishop(1,6) bishop(7,6) bishop(1,7) bishop(3,7) bishop(5,7) bishop(7,7) Optimization: -11 Answer: 13 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) Optimization: -12 OPTIMUM FOUND Models: 13 Optimum: yes Optimization: -12 Calls: 1 Time: 0.825s (Solving: 0.82s 1st Model: 0.00s Unsat: 0.78s) CPU Time: 0.766s **For n = 8 (8 ranks, 8 files)** clingo version 5.4.0 Reading from pa1_8.txt Solving... Answer: 1 Optimization: 0 Answer: 2 bishop(7,8) Optimization: -1 Answer: 3 bishop(3,1) bishop(7,8) Optimization: -2 Answer: 4 bishop(3,1) bishop(2,8) bishop(7,8) Optimization: -3 Answer: 5 bishop(3,1) bishop(8,5) bishop(2,8) bishop(7,8) Optimization: -4 Answer: 6 bishop(3,1) bishop(1,5) bishop(8,5) bishop(2,8) bishop(7,8) Optimization: -5 Answer: 7 bishop(1,3) bishop(1,5) bishop(8,5) bishop(1,7) bishop(8,7) bishop(3,8) Optimization: -6 Answer: 8 bishop(3,1) bishop(4,1) bishop(8,1) bishop(1,5) bishop(8,7) bishop(2,8) bishop(3,8)

	Optimization: -7 Answer: 9 bishop(3,1) bishop(6,1) bishop(8,1) bishop(1,5) bishop(8,5) bishop(1,7) bishop(8,7) bishop(3,8) Optimization: -8 Answer: 10 bishop(4,1) bishop(6,1) bishop(8,1) bishop(1,3) bishop(8,4) bishop(1,5) bishop(8,7) bishop(2,8) bishop(3,8) Optimization: -9 Answer: 11 bishop(3,1) bishop(6,1) bishop(8,1) bishop(8,2) bishop(8,4) bishop(1,5) bishop(8,5) bishop(1,7) bishop(3,8) bishop(7,8) Optimization: -10 Answer: 12 bishop(2,1) bishop(7,3) bishop(8,3) bishop(2,4) bishop(8,5) bishop(1,6) bishop(2,6) bishop(3,6) bishop(6,6) bishop(8,6) bishop(7,8) Optimization: -11 Answer: 13 bishop(2,1) bishop(4,1) bishop(5,1) bishop(6,1) bishop(7,1) bishop(8,1) bishop(2,2) bishop(8,6) bishop(5,7) bishop(6,7) bishop(2,8) bishop(3,8) Optimization: -12 Answer: 14 bishop(2,1) bishop(4,1) bishop(5,1) bishop(6,1) bishop(7,1) bishop(8,1) bishop(2,2) bishop(8,6) bishop(5,7) bishop(2,8) bishop(3,8) bishop(5,8) bishop(7,8) Optimization: -13 Answer: 15 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) Optimization: -14 OPTIMUM FOUND Models: 15 Optimum: yes Optimization: -14 Calls: 1 Time: 36.969s (Solving: 36.96s 1st Model: 0.00s Unsat: 36.49s) CPU Time : 34.891s
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) = 2(n-1)

Problem 9: About a set X of numbers, we say that it is almost sum-free if the sum of two different elements of X never belongs to X. For instance, the set {1, 2, 4} is almost sum-free. Almost-Schur number A(k) is the largest integer n for which the interval {1, . . . , n} can be partitioned into k almost sum-free sets.

Use clingo to find the exact values of A(1), A(2), A(3) and try to find the largest lower bound for A(4), i.e., the largest number l such that A(4) ≥ l.
***Hint*: you do not need to find all partitions to find the values of A(k).**

Input Program	% Assign each number from 1 to maxNum to exactly one of k groups 1 {assignNumToGroup(Num,1..k)} = 1 :- Num = 1..n. % Prevent any two numbers from being in the same group if their sum also belongs to that group :- assignNumToGroup(First,Group), assignNumToGroup(Second,Group), assignNumToGroup(First+Second,Group), First!=Second.
Command Line	You should write multiple command lines below. clingo -c k=1 -c n=2 pa1_9.txt 0 clingo -c k=2 -c n=8 pa1_9.txt 0 clingo -c k=3 -c n=23 pa1_9.txt 0 clingo -c k=4 -c n=66 pa1_9.txt 0
Output of clingo	**k=1 and n= 2** clingo version 5.4.0 Reading from pa1_9.txt Solving... Answer: 1 assignNumToGroup(1,1) assignNumToGroup(2,1) SATISFIABLE Models : 1 Calls : 1 Time : 0.007s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s) CPU Time : 0.000s **k=2 and n= 8** clingo version 5.4.0 Reading from pa1_9.txt Solving... Answer: 1

	assignNumToGroup(1,1) assignNumToGroup(2,1) assignNumToGroup(3,2) assignNumToGroup(4,1) assignNumToGroup(5,2) assignNumToGroup(6,2) assignNumToGroup(7,2) assignNumToGroup(8,1) Answer: 2 assignNumToGroup(3,1) assignNumToGroup(1,2) assignNumToGroup(2,2) assignNumToGroup(4,2) assignNumToGroup(5,1) assignNumToGroup(6,1) assignNumToGroup(7,1) assignNumToGroup(8,2) SATISFIABLE Models: 2 Calls: 1 Time: 0.009s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s) CPU Time: 0.000s
	k = 3 and n = 23 clingo version 5.4.0 Reading from pa1_9.txt Solving... Answer: 1 assignNumToGroup(1,2) assignNumToGroup(2,2) assignNumToGroup(3,3) assignNumToGroup(4,2) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,2) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,2) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,2) assignNumToGroup(17,1) assignNumToGroup(18,1) assignNumToGroup(19,3) assignNumToGroup(20,1) assignNumToGroup(21,3) assignNumToGroup(22,2) assignNumToGroup(23,3) Answer: 2 assignNumToGroup(1,2) assignNumToGroup(2,2) assignNumToGroup(3,3) assignNumToGroup(4,2) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,2) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,2) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,1) assignNumToGroup(17,1) assignNumToGroup(18,1) assignNumToGroup(19,3) assignNumToGroup(20,1) assignNumToGroup(21,3) assignNumToGroup(22,2) assignNumToGroup(23,3) Answer: 3 assignNumToGroup(1,2) assignNumToGroup(2,2) assignNumToGroup(3,3) assignNumToGroup(4,2) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,2) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,2)

	assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,1) assignNumToGroup(17,2) assignNumToGroup(18,1) assignNumToGroup(19,3) assignNumToGroup(20,1) assignNumToGroup(21,3) assignNumToGroup(22,2) assignNumToGroup(23,3) Answer: 4 assignNumToGroup(3,2) assignNumToGroup(1,3) assignNumToGroup(2,3) assignNumToGroup(4,3) assignNumToGroup(5,2) assignNumToGroup(6,2) assignNumToGroup(7,2) assignNumToGroup(8,3) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,3) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,1) assignNumToGroup(17,1) assignNumToGroup(18,1) assignNumToGroup(19,2) assignNumToGroup(20,1) assignNumToGroup(21,2) assignNumToGroup(22,3) assignNumToGroup(23,2) Answer: 5 assignNumToGroup(3,2) assignNumToGroup(1,3) assignNumToGroup(2,3) assignNumToGroup(4,3) assignNumToGroup(5,2) assignNumToGroup(6,2) assignNumToGroup(7,2) assignNumToGroup(8,3) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,3) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,1) assignNumToGroup(17,3) assignNumToGroup(18,1) assignNumToGroup(19,2) assignNumToGroup(20,1) assignNumToGroup(21,2) assignNumToGroup(22,3) assignNumToGroup(23,2) Answer: 6 assignNumToGroup(3,2) assignNumToGroup(1,3) assignNumToGroup(2,3) assignNumToGroup(4,3) assignNumToGroup(5,2) assignNumToGroup(6,2) assignNumToGroup(7,2) assignNumToGroup(8,3) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,3) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,3) assignNumToGroup(17,1) assignNumToGroup(18,1) assignNumToGroup(19,2) assignNumToGroup(20,1) assignNumToGroup(21,2) assignNumToGroup(22,3) assignNumToGroup(23,2) Answer: 7 assignNumToGroup(1,1) assignNumToGroup(2,1) assignNumToGroup(3,3) assignNumToGroup(4,1) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,1) assignNumToGroup(9,2) assignNumToGroup(10,2) assignNumToGroup(11,1)

	assignNumToGroup(12,2) assignNumToGroup(13,2) assignNumToGroup(14,2) assignNumToGroup(15,2) assignNumToGroup(16,1) assignNumToGroup(17,2) assignNumToGroup(18,2) assignNumToGroup(19,3) assignNumToGroup(20,2) assignNumToGroup(21,3) assignNumToGroup(22,1) assignNumToGroup(23,3) Answer: 8 assignNumToGroup(1,1) assignNumToGroup(2,1) assignNumToGroup(3,3) assignNumToGroup(4,1) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,1) assignNumToGroup(9,2) assignNumToGroup(10,2) assignNumToGroup(11,1) assignNumToGroup(12,2) assignNumToGroup(13,2) assignNumToGroup(14,2) assignNumToGroup(15,2) assignNumToGroup(16,2) assignNumToGroup(17,1) assignNumToGroup(18,2) assignNumToGroup(19,3) assignNumToGroup(20,2) assignNumToGroup(21,3) assignNumToGroup(22,1) assignNumToGroup(23,3) Answer: 9 assignNumToGroup(1,1) assignNumToGroup(2,1) assignNumToGroup(3,3) assignNumToGroup(4,1) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,1) assignNumToGroup(9,2) assignNumToGroup(10,2) assignNumToGroup(11,1) assignNumToGroup(12,2) assignNumToGroup(13,2) assignNumToGroup(14,2) assignNumToGroup(15,2) assignNumToGroup(16,2) assignNumToGroup(17,2) assignNumToGroup(18,2) assignNumToGroup(19,3) assignNumToGroup(20,2) assignNumToGroup(21,3) assignNumToGroup(22,1) assignNumToGroup(23,3) Answer: 10 assignNumToGroup(1,1) assignNumToGroup(2,1) assignNumToGroup(3,2) assignNumToGroup(4,1) assignNumToGroup(5,2) assignNumToGroup(6,2) assignNumToGroup(7,2) assignNumToGroup(8,1) assignNumToGroup(9,3) assignNumToGroup(10,3) assignNumToGroup(11,1) assignNumToGroup(12,3) assignNumToGroup(13,3) assignNumToGroup(14,3) assignNumToGroup(15,3) assignNumToGroup(16,1) assignNumToGroup(17,3) assignNumToGroup(18,3) assignNumToGroup(19,2) assignNumToGroup(20,3) assignNumToGroup(21,2) assignNumToGroup(22,1) assignNumToGroup(23,2) Answer: 11 assignNumToGroup(1,1) assignNumToGroup(2,1) assignNumToGroup(3,2) assignNumToGroup(4,1) assignNumToGroup(5,2) assignNumToGroup(6,2) assignNumToGroup(7,2) assignNumToGroup(8,1) assignNumToGroup(9,3) assignNumToGroup(10,3) assignNumToGroup(11,1)

	assignNumToGroup(12,3) assignNumToGroup(13,3) assignNumToGroup(14,3) assignNumToGroup(15,3) assignNumToGroup(16,3) assignNumToGroup(17,1) assignNumToGroup(18,3) assignNumToGroup(19,2) assignNumToGroup(20,3) assignNumToGroup(21,2) assignNumToGroup(22,1) assignNumToGroup(23,2) Answer: 12 assignNumToGroup(1,1) assignNumToGroup(2,1) assignNumToGroup(3,2) assignNumToGroup(4,1) assignNumToGroup(5,2) assignNumToGroup(6,2) assignNumToGroup(7,2) assignNumToGroup(8,1) assignNumToGroup(9,3) assignNumToGroup(10,3) assignNumToGroup(11,1) assignNumToGroup(12,3) assignNumToGroup(13,3) assignNumToGroup(14,3) assignNumToGroup(15,3) assignNumToGroup(16,3) assignNumToGroup(17,3) assignNumToGroup(18,3) assignNumToGroup(19,2) assignNumToGroup(20,3) assignNumToGroup(21,2) assignNumToGroup(22,1) assignNumToGroup(23,2) Answer: 13 assignNumToGroup(3,1) assignNumToGroup(1,2) assignNumToGroup(2,2) assignNumToGroup(4,2) assignNumToGroup(5,1) assignNumToGroup(6,1) assignNumToGroup(7,1) assignNumToGroup(8,2) assignNumToGroup(9,3) assignNumToGroup(10,3) assignNumToGroup(11,2) assignNumToGroup(12,3) assignNumToGroup(13,3) assignNumToGroup(14,3) assignNumToGroup(15,3) assignNumToGroup(16,3) assignNumToGroup(17,3) assignNumToGroup(18,3) assignNumToGroup(19,1) assignNumToGroup(20,3) assignNumToGroup(21,1) assignNumToGroup(22,2) assignNumToGroup(23,1) Answer: 14 assignNumToGroup(3,1) assignNumToGroup(1,2) assignNumToGroup(2,2) assignNumToGroup(4,2) assignNumToGroup(5,1) assignNumToGroup(6,1) assignNumToGroup(7,1) assignNumToGroup(8,2) assignNumToGroup(9,3) assignNumToGroup(10,3) assignNumToGroup(11,2) assignNumToGroup(12,3) assignNumToGroup(13,3) assignNumToGroup(14,3) assignNumToGroup(15,3) assignNumToGroup(16,2) assignNumToGroup(17,3) assignNumToGroup(18,3) assignNumToGroup(19,1) assignNumToGroup(20,3) assignNumToGroup(21,1) assignNumToGroup(22,2) assignNumToGroup(23,1) Answer: 15 assignNumToGroup(3,1) assignNumToGroup(1,2) assignNumToGroup(2,2) assignNumToGroup(4,2) assignNumToGroup(5,1) assignNumToGroup(6,1) assignNumToGroup(7,1) assignNumToGroup(8,2) assignNumToGroup(9,3) assignNumToGroup(10,3) assignNumToGroup(11,2)

	assignNumToGroup(12,3) assignNumToGroup(13,3) assignNumToGroup(14,3) assignNumToGroup(15,3) assignNumToGroup(16,3) assignNumToGroup(17,2) assignNumToGroup(18,3) assignNumToGroup(19,1) assignNumToGroup(20,3) assignNumToGroup(21,1) assignNumToGroup(22,2) assignNumToGroup(23,1) Answer: 16 assignNumToGroup(3,1) assignNumToGroup(1,3) assignNumToGroup(2,3) assignNumToGroup(4,3) assignNumToGroup(5,1) assignNumToGroup(6,1) assignNumToGroup(7,1) assignNumToGroup(8,3) assignNumToGroup(9,2) assignNumToGroup(10,2) assignNumToGroup(11,3) assignNumToGroup(12,2) assignNumToGroup(13,2) assignNumToGroup(14,2) assignNumToGroup(15,2) assignNumToGroup(16,3) assignNumToGroup(17,2) assignNumToGroup(18,2) assignNumToGroup(19,1) assignNumToGroup(20,2) assignNumToGroup(21,1) assignNumToGroup(22,3) assignNumToGroup(23,1) Answer: 17 assignNumToGroup(3,1) assignNumToGroup(1,3) assignNumToGroup(2,3) assignNumToGroup(4,3) assignNumToGroup(5,1) assignNumToGroup(6,1) assignNumToGroup(7,1) assignNumToGroup(8,3) assignNumToGroup(9,2) assignNumToGroup(10,2) assignNumToGroup(11,3) assignNumToGroup(12,2) assignNumToGroup(13,2) assignNumToGroup(14,2) assignNumToGroup(15,2) assignNumToGroup(16,2) assignNumToGroup(17,2) assignNumToGroup(18,2) assignNumToGroup(19,1) assignNumToGroup(20,2) assignNumToGroup(21,1) assignNumToGroup(22,3) assignNumToGroup(23,1) Answer: 18 assignNumToGroup(3,1) assignNumToGroup(1,3) assignNumToGroup(2,3) assignNumToGroup(4,3) assignNumToGroup(5,1) assignNumToGroup(6,1) assignNumToGroup(7,1) assignNumToGroup(8,3) assignNumToGroup(9,2) assignNumToGroup(10,2) assignNumToGroup(11,3) assignNumToGroup(12,2) assignNumToGroup(13,2) assignNumToGroup(14,2) assignNumToGroup(15,2) assignNumToGroup(16,2) assignNumToGroup(17,3) assignNumToGroup(18,2) assignNumToGroup(19,1) assignNumToGroup(20,2) assignNumToGroup(21,1) assignNumToGroup(22,3) assignNumToGroup(23,1) SATISFIABLE Models : 18 Calls : 1 Time : 0.154s (Solving: 0.14s 1st Model: 0.01s Unsat: 0.01s)

	CPU Time : 0.063s **k = 4 and n = 66** clingo version 5.4.0 Reading from pa1_9.txt Solving... Answer: 29927 assignNumToGroup(3,3) assignNumToGroup(1,4) assignNumToGroup(2,4) assignNumToGroup(4,4) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,4) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,4) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,1) assignNumToGroup(17,1) assignNumToGroup(18,1) assignNumToGroup(19,3) assignNumToGroup(20,1) assignNumToGroup(21,3) assignNumToGroup(22,4) assignNumToGroup(23,3) assignNumToGroup(24,2) assignNumToGroup(25,4) assignNumToGroup(26,2) assignNumToGroup(27,2) assignNumToGroup(28,2) assignNumToGroup(29,2) assignNumToGroup(30,2) assignNumToGroup(31,2) assignNumToGroup(32,4) assignNumToGroup(33,2) assignNumToGroup(34,2) assignNumToGroup(35,4) assignNumToGroup(36,2) assignNumToGroup(37,3) assignNumToGroup(38,3) assignNumToGroup(39,3) assignNumToGroup(40,2) assignNumToGroup(41,2) assignNumToGroup(42,2) assignNumToGroup(43,2) assignNumToGroup(44,2) assignNumToGroup(45,2) assignNumToGroup(46,2) assignNumToGroup(47,2) assignNumToGroup(48,2) assignNumToGroup(49,2) assignNumToGroup(50,4) assignNumToGroup(51,3) assignNumToGroup(52,3) assignNumToGroup(53,3) assignNumToGroup(54,1) assignNumToGroup(55,1) assignNumToGroup(56,1) assignNumToGroup(57,1) assignNumToGroup(58,1) assignNumToGroup(59,4) assignNumToGroup(60,1) assignNumToGroup(61,1) assignNumToGroup(62,1) assignNumToGroup(63,3) assignNumToGroup(64,3) assignNumToGroup(65,3) assignNumToGroup(66,4) Answer: 29928 assignNumToGroup(3,3) assignNumToGroup(1,4) assignNumToGroup(2,4) assignNumToGroup(4,4)

	assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,4) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,4) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,1) assignNumToGroup(17,1) assignNumToGroup(18,1) assignNumToGroup(19,3) assignNumToGroup(20,1) assignNumToGroup(21,3) assignNumToGroup(22,4) assignNumToGroup(23,3) assignNumToGroup(24,2) assignNumToGroup(25,4) assignNumToGroup(26,2) assignNumToGroup(27,2) assignNumToGroup(28,2) assignNumToGroup(29,2) assignNumToGroup(30,2) assignNumToGroup(31,2) assignNumToGroup(32,4) assignNumToGroup(33,2) assignNumToGroup(34,2) assignNumToGroup(35,4) assignNumToGroup(36,2) assignNumToGroup(37,3) assignNumToGroup(38,4) assignNumToGroup(39,3) assignNumToGroup(40,2) assignNumToGroup(41,2) assignNumToGroup(42,2) assignNumToGroup(43,2) assignNumToGroup(44,2) assignNumToGroup(45,4) assignNumToGroup(46,2) assignNumToGroup(47,2) assignNumToGroup(48,2) assignNumToGroup(49,2) assignNumToGroup(50,4) assignNumToGroup(51,3) assignNumToGroup(52,3) assignNumToGroup(53,3) assignNumToGroup(54,1) assignNumToGroup(55,1) assignNumToGroup(56,1) assignNumToGroup(57,1) assignNumToGroup(58,1) assignNumToGroup(59,1) assignNumToGroup(60,1) assignNumToGroup(61,1) assignNumToGroup(62,1) assignNumToGroup(63,3) assignNumToGroup(64,3) assignNumToGroup(65,3) assignNumToGroup(66,4) Answer: 29929 assignNumToGroup(3,3) assignNumToGroup(1,4) assignNumToGroup(2,4) assignNumToGroup(4,4) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,4) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,4) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,1) assignNumToGroup(17,1) assignNumToGroup(18,1) assignNumToGroup(19,3) assignNumToGroup(20,1) assignNumToGroup(21,3) assignNumToGroup(22,4)

	assignNumToGroup(23,3) assignNumToGroup(24,2) assignNumToGroup(25,4) assignNumToGroup(26,2) assignNumToGroup(27,2) assignNumToGroup(28,2) assignNumToGroup(29,2) assignNumToGroup(30,2) assignNumToGroup(31,2) assignNumToGroup(32,4) assignNumToGroup(33,2) assignNumToGroup(34,2) assignNumToGroup(35,4) assignNumToGroup(36,2) assignNumToGroup(37,3) assignNumToGroup(38,4) assignNumToGroup(39,3) assignNumToGroup(40,2) assignNumToGroup(41,2) assignNumToGroup(42,2) assignNumToGroup(43,2) assignNumToGroup(44,2) assignNumToGroup(45,4) assignNumToGroup(46,2) assignNumToGroup(47,2) assignNumToGroup(48,2) assignNumToGroup(49,2) assignNumToGroup(50,4) assignNumToGroup(51,3) assignNumToGroup(52,3) assignNumToGroup(53,3) assignNumToGroup(54,1) assignNumToGroup(55,1) assignNumToGroup(56,1) assignNumToGroup(57,1) assignNumToGroup(58,1) assignNumToGroup(59,4) assignNumToGroup(60,1) assignNumToGroup(61,1) assignNumToGroup(62,1) assignNumToGroup(63,3) assignNumToGroup(64,3) assignNumToGroup(65,3) assignNumToGroup(66,4) Answer: 29930 assignNumToGroup(3,3) assignNumToGroup(1,4) assignNumToGroup(2,4) assignNumToGroup(4,4) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,4) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,4) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,1) assignNumToGroup(17,1) assignNumToGroup(18,1) assignNumToGroup(19,3) assignNumToGroup(20,1) assignNumToGroup(21,3) assignNumToGroup(22,4) assignNumToGroup(23,3) assignNumToGroup(24,2) assignNumToGroup(25,4) assignNumToGroup(26,2) assignNumToGroup(27,2) assignNumToGroup(28,2) assignNumToGroup(29,2) assignNumToGroup(30,2) assignNumToGroup(31,2) assignNumToGroup(32,4) assignNumToGroup(33,2) assignNumToGroup(34,2) assignNumToGroup(35,4) assignNumToGroup(36,2) assignNumToGroup(37,3) assignNumToGroup(38,3) assignNumToGroup(39,3) assignNumToGroup(40,2)

	assignNumToGroup(41,2) assignNumToGroup(42,2) assignNumToGroup(43,2) assignNumToGroup(44,2) assignNumToGroup(45,4) assignNumToGroup(46,2) assignNumToGroup(47,2) assignNumToGroup(48,2) assignNumToGroup(49,2) assignNumToGroup(50,4) assignNumToGroup(51,3) assignNumToGroup(52,3) assignNumToGroup(53,3) assignNumToGroup(54,1) assignNumToGroup(55,1) assignNumToGroup(56,1) assignNumToGroup(57,1) assignNumToGroup(58,1) assignNumToGroup(59,1) assignNumToGroup(60,1) assignNumToGroup(61,1) assignNumToGroup(62,1) assignNumToGroup(63,3) assignNumToGroup(64,3) assignNumToGroup(65,3) assignNumToGroup(66,4) Answer: 29931 assignNumToGroup(3,3) assignNumToGroup(1,4) assignNumToGroup(2,4) assignNumToGroup(4,4) assignNumToGroup(5,3) assignNumToGroup(6,3) assignNumToGroup(7,3) assignNumToGroup(8,4) assignNumToGroup(9,1) assignNumToGroup(10,1) assignNumToGroup(11,4) assignNumToGroup(12,1) assignNumToGroup(13,1) assignNumToGroup(14,1) assignNumToGroup(15,1) assignNumToGroup(16,1) assignNumToGroup(17,1) assignNumToGroup(18,1) assignNumToGroup(19,3) assignNumToGroup(20,1) assignNumToGroup(21,3) assignNumToGroup(22,4) assignNumToGroup(23,3) assignNumToGroup(24,2) assignNumToGroup(25,4) assignNumToGroup(26,2) assignNumToGroup(27,2) assignNumToGroup(28,2) assignNumToGroup(29,2) assignNumToGroup(30,2) assignNumToGroup(31,2) assignNumToGroup(32,4) assignNumToGroup(33,2) assignNumToGroup(34,2) assignNumToGroup(35,4) assignNumToGroup(36,2) assignNumToGroup(37,3) assignNumToGroup(38,3) assignNumToGroup(39,3) assignNumToGroup(40,2) assignNumToGroup(41,2) assignNumToGroup(42,2) assignNumToGroup(43,2) assignNumToGroup(44,2) assignNumToGroup(45,4) assignNumToGroup(46,2) assignNumToGroup(47,2) assignNumToGroup(48,2) assignNumToGroup(49,2) assignNumToGroup(50,4) assignNumToGroup(51,3) assignNumToGroup(52,3) assignNumToGroup(53,3) assignNumToGroup(54,1) assignNumToGroup(55,1) assignNumToGroup(56,1) assignNumToGroup(57,1) assignNumToGroup(58,1)

<table>
<tr><td></td><td>assignNumToGroup(59,4) assignNumToGroup(60,1)
assignNumToGroup(61,1) assignNumToGroup(62,1)
assignNumToGroup(63,3) assignNumToGroup(64,3)
assignNumToGroup(65,3) assignNumToGroup(66,4)
^C*** Info : (clingo): INTERRUPTED by signal!
SATISFIABLE

INTERRUPTED: 1
Models: 29931+
Calls: 1
Time: 2263.925s (Solving: 2263.87s 1st Model: 181.88s Unsat: 0.00s)
CPU Time: 1212.448s</td></tr>
<tr><td>Answer
to Questions</td><td>Fill in the values accordingly.

<table><tr><td>Exact value of A(1)</td><td>2</td></tr><tr><td>Exact value of A(2)</td><td>8</td></tr><tr><td>Exact value of A(3)</td><td>23</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>66</td></tr></table></td></tr>
</table>