**CSE 579**

**Programming Assignment 1**

**CLINGO PROGRAMS**

Problem 1

|  |  |
| --- | --- |
| Input  Program | % 10\*10 board size queens problem.  #const n = 10.  % 1. Create a n \* n board  {n\_queens\_soln(R, 1..n)}=1:-R=1..n.  % 2. Queens not in the same row.  :- n\_queens\_soln(R1, C), n\_queens\_soln(R2, C), R1 != R2.  % 3. Queens not in the same column.  :- n\_queens\_soln(R, C1), n\_queens\_soln(R, C2), C1 != C2.  % 4. No 2 queens in same diagonal  :- n\_queens\_soln(R1, C1), n\_queens\_soln(R2, C2), R1 != R2, |R1 - R2| == |C1 - C2|.  % 5. No queens in the 4 \* 4 = 16 squares in the 4 to 7 rows and columns of the board  :- n\_queens\_soln(R, C), R >= 4, R <= 7, C >= 4, C <= 7. |
| Command  Line | clingo.exe .\P1\_NQueensMod\_Q1.lp 0 |
| Output  of clingo | clingo version 5.6.2  Reading from .\P1\_NQueensMod\_Q1.lp  Solving...  Answer: 1  n\_queens\_soln(2,1) n\_queens\_soln(9,10) n\_queens\_soln(7,9) n\_queens\_soln(5,8) n\_queens\_soln(3,7) n\_queens\_soln(4,2) n\_queens\_soln(1,6) n\_queens\_soln(6,3) n\_queens\_soln(8,4) n\_queens\_soln(10,5)  Answer: 2  n\_queens\_soln(7,8) n\_queens\_soln(3,5) n\_queens\_soln(5,1) n\_queens\_soln(6,10) n\_queens\_soln(2,7) n\_queens\_soln(4,3) n\_queens\_soln(1,9) n\_queens\_soln(10,2) n\_queens\_soln(8,6) n\_queens\_soln(9,4)  Answer: 3  n\_queens\_soln(3,4) n\_queens\_soln(6,8) n\_queens\_soln(1,5) n\_queens\_soln(4,9) n\_queens\_soln(5,3) n\_queens\_soln(7,2) n\_queens\_soln(2,10) n\_queens\_soln(9,1) n\_queens\_soln(8,7) n\_queens\_soln(10,6)  Answer: 4  n\_queens\_soln(1,2) n\_queens\_soln(2,4) n\_queens\_soln(3,6) n\_queens\_soln(4,8) n\_queens\_soln(5,10) n\_queens\_soln(6,1) n\_queens\_soln(7,3) n\_queens\_soln(10,9) n\_queens\_soln(9,7) n\_queens\_soln(8,5)  Answer: 5  n\_queens\_soln(9,10) n\_queens\_soln(5,8) n\_queens\_soln(3,7) n\_queens\_soln(4,2) n\_queens\_soln(1,6) n\_queens\_soln(7,1) n\_queens\_soln(2,9) n\_queens\_soln(6,3) n\_queens\_soln(8,4) n\_queens\_soln(10,5)  Answer: 6  n\_queens\_soln(1,2) n\_queens\_soln(3,6) n\_queens\_soln(4,8) n\_queens\_soln(5,10) n\_queens\_soln(6,1) n\_queens\_soln(2,9) n\_queens\_soln(7,3) n\_queens\_soln(9,7) n\_queens\_soln(8,5) n\_queens\_soln(10,4)  Answer: 7  n\_queens\_soln(2,2) n\_queens\_soln(7,8) n\_queens\_soln(3,5) n\_queens\_soln(5,1) n\_queens\_soln(6,10) n\_queens\_soln(4,3) n\_queens\_soln(1,9) n\_queens\_soln(8,6) n\_queens\_soln(9,4) n\_queens\_soln(10,7)  <TRUNCATED>  n\_queens\_soln(3,4) n\_queens\_soln(7,9) n\_queens\_soln(4,1) n\_queens\_soln(5,8) n\_queens\_soln(2,7) n\_queens\_soln(6,2) n\_queens\_soln(1,10) n\_queens\_soln(9,3) n\_queens\_soln(8,6) n\_queens\_soln(10,5)  Answer: 35  n\_queens\_soln(3,5) n\_queens\_soln(1,4) n\_queens\_soln(5,1) n\_queens\_soln(6,10) n\_queens\_soln(4,9) n\_queens\_soln(2,8) n\_queens\_soln(7,2) n\_queens\_soln(9,3) n\_queens\_soln(8,6) n\_queens\_soln(10,7)  Answer: 36  n\_queens\_soln(2,2) n\_queens\_soln(7,8) n\_queens\_soln(3,5) n\_queens\_soln(1,4) n\_queens\_soln(5,1) n\_queens\_soln(6,10) n\_queens\_soln(4,9) n\_queens\_soln(9,3) n\_queens\_soln(8,6) n\_queens\_soln(10,7)  Answer: 37  n\_queens\_soln(2,3) n\_queens\_soln(7,9) n\_queens\_soln(3,6) n\_queens\_soln(4,2) n\_queens\_soln(5,10) n\_queens\_soln(6,1) n\_queens\_soln(1,7) n\_queens\_soln(9,8) n\_queens\_soln(8,5) n\_queens\_soln(10,4)  Answer: 38  n\_queens\_soln(3,6) n\_queens\_soln(4,2) n\_queens\_soln(5,10) n\_queens\_soln(6,1) n\_queens\_soln(1,7) n\_queens\_soln(2,9) n\_queens\_soln(7,3) n\_queens\_soln(9,8) n\_queens\_soln(8,5) n\_queens\_soln(10,4)  Answer: 39  n\_queens\_soln(2,4) n\_queens\_soln(6,9) n\_queens\_soln(3,7) n\_queens\_soln(1,6) n\_queens\_soln(4,10) n\_queens\_soln(5,3) n\_queens\_soln(7,2) n\_queens\_soln(10,1) n\_queens\_soln(9,8) n\_queens\_soln(8,5)  Answer: 40  n\_queens\_soln(1,1) n\_queens\_soln(2,4) n\_queens\_soln(6,9) n\_queens\_soln(3,7) n\_queens\_soln(4,10) n\_queens\_soln(5,3) n\_queens\_soln(7,2) n\_queens\_soln(9,8) n\_queens\_soln(8,5) n\_queens\_soln(10,6)  SATISFIABLE  Models : 40  Calls : 1  Time : 0.044s (Solving: 0.03s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.000s |

Problem 2

|  |  |
| --- | --- |
| Input  Program | % 1. Create a n \* n board  {n\_quins\_sizeN(R, 1..n)}=1 :- R=1..n.  % 2. Queens not in the same row.  :- n\_quins\_sizeN (R1, C), n\_quins\_sizeN (R2, C), R1 != R2.  % 3. Queens not in the same column.  :- n\_quins\_sizeN (R, C1), n\_quins\_sizeN (R, C2), C1 != C2.  % 4. No 2 Queens in same diagonal  :- n\_quins\_sizeN (R1, C1), n\_quins\_sizeN (R2, C2), R1 != R2, |R1-R2|=|C1-C2|. |
| Command  Line | You should write multiple command lines below.   1. clingo -c n=3 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 2. clingo -c n=4 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 3. clingo -c n=5 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 4. clingo -c n=6 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 5. clingo -c n=7 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 6. clingo -c n=8 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 7. clingo -c n=9 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 8. clingo -c n=10 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 9. clingo -c n=11 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 10. clingo -c n=12 .\P1\_NQueensProblem\_Comp\_Q2.lp 0 |
| Output  of clingo | Since the output is large, do not copy them into the submission. |
| 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 | CPU Time | Time | | 3 | 0 | 0.000s | 0.005s | | 4 | 2 | 0.000s | 0.000s | | 5 | 10 | 0.000s | 0.011s | | 6 | 4 | 0.000s | 0.007s | | 7 | 40 | 0.000s | 0.022s | | 8 | 92 | 0.000s | 0.049s | | 9 | 352 | 0.000s | 0.191s | | 10 | 724 | 0.000s | 0.540s | | 11 | 2680 | 0.109s | 3.310s | | 12 | 14200 | 5.281s | 50.937s | |

Problem 3

|  |  |
| --- | --- |
| Input  Program | % First create a grid space which allows numbers from 1 to 9  % in a simple\_sudoku\_solver grid of size 9  1{ simple\_sudoku\_solver(X, Y, N): X=1..9, Y=1..9, X>=X1, X<=X1+2, Y>=Y1, Y<=Y1+2 }1 :- N=1..9, X1=3\*(0..2)+1, Y1=3\*(0..2)+1.  % Secondly, no 2 numbers are different in the same row and column  :- simple\_sudoku\_solver(X, Y, N), simple\_sudoku\_solver(X, Y, N1), N!=N1.  % Thirdly, no 2 columns are different given a row and number  :- simple\_sudoku\_solver(X, Y, N), simple\_sudoku\_solver(X, Y1, N), Y!=Y1.  % fourthly no 2 rows are different given a column and number  :- simple\_sudoku\_solver(X, Y, N), simple\_sudoku\_solver(X1, Y, N), X!=X1.  % box 1  simple\_sudoku\_solver(1, 1, 1).  simple\_sudoku\_solver(2, 2, 3).  simple\_sudoku\_solver(3, 3, 9).  % box 2  simple\_sudoku\_solver(4, 3, 5).  simple\_sudoku\_solver(5, 2, 1).  simple\_sudoku\_solver(6, 1, 6).  % box 3  simple\_sudoku\_solver(7, 1, 3).  simple\_sudoku\_solver(8, 2, 4).  simple\_sudoku\_solver(9, 3, 7).  % box 4  simple\_sudoku\_solver(1, 6, 7).  simple\_sudoku\_solver(2, 5, 2).  simple\_sudoku\_solver(3, 4, 6).  % box 5  simple\_sudoku\_solver(4, 4, 3).  simple\_sudoku\_solver(5, 5, 8).  simple\_sudoku\_solver(6, 6, 4).  % box 6  % box 7  simple\_sudoku\_solver(1, 8, 9).  simple\_sudoku\_solver(2, 9, 8).  simple\_sudoku\_solver(3, 7, 5).  % box 8  simple\_sudoku\_solver(4, 7, 9).  simple\_sudoku\_solver(5, 9, 2).  % box 9  simple\_sudoku\_solver(7, 8, 1).  simple\_sudoku\_solver(8, 9, 7).  simple\_sudoku\_solver(9, 7, 3). |
| Command  Line | clingo .\P1\_Sudoku\_Solver\_Q3.lp 0 |
| Output  of clingo | clingo version 5.6.2  Reading from .\P1\_Sudoku\_Solver\_Q3.lp  Solving...  Answer: 1  simple\_sudoku\_solver(1,1,1) simple\_sudoku\_solver(2,2,3) simple\_sudoku\_solver(3,3,9) simple\_sudoku\_solver(4,3,5) simple\_sudoku\_solver(5,2,1) simple\_sudoku\_solver(6,1,6) simple\_sudoku\_solver(7,1,3) simple\_sudoku\_solver(8,2,4) simple\_sudoku\_solver(9,3,7) simple\_sudoku\_solver(1,6,7) simple\_sudoku\_solver(2,5,2) simple\_sudoku\_solver(3,4,6) simple\_sudoku\_solver(4,4,3) simple\_sudoku\_solver(5,5,8) simple\_sudoku\_solver(6,6,4) simple\_sudoku\_solver(1,8,9) simple\_sudoku\_solver(2,9,8) simple\_sudoku\_solver(3,7,5) simple\_sudoku\_solver(4,7,9) simple\_sudoku\_solver(5,9,2) simple\_sudoku\_solver(7,8,1) simple\_sudoku\_solver(8,9,7) simple\_sudoku\_solver(9,7,3) simple\_sudoku\_solver(8,3,1) simple\_sudoku\_solver(8,1,2) simple\_sudoku\_solver(6,2,2) simple\_sudoku\_solver(1,3,2) simple\_sudoku\_solver(5,3,3) simple\_sudoku\_solver(4,1,4) simple\_sudoku\_solver(2,3,4) simple\_sudoku\_solver(2,1,5) simple\_sudoku\_solver(7,2,5) simple\_sudoku\_solver(1,2,6) simple\_sudoku\_solver(7,3,6) simple\_sudoku\_solver(3,1,7) simple\_sudoku\_solver(4,2,7) simple\_sudoku\_solver(9,1,8) simple\_sudoku\_solver(3,2,8) simple\_sudoku\_solver(6,3,8) simple\_sudoku\_solver(5,1,9) simple\_sudoku\_solver(9,2,9) simple\_sudoku\_solver(2,4,1) simple\_sudoku\_solver(4,5,1) simple\_sudoku\_solver(9,6,1) simple\_sudoku\_solver(9,4,2) simple\_sudoku\_solver(4,6,2) simple\_sudoku\_solver(8,5,3) simple\_sudoku\_solver(3,6,3) simple\_sudoku\_solver(7,4,4) simple\_sudoku\_solver(3,5,4) simple\_sudoku\_solver(5,4,5) simple\_sudoku\_solver(1,5,5) simple\_sudoku\_solver(8,6,5) simple\_sudoku\_solver(9,5,6) simple\_sudoku\_solver(5,6,6) simple\_sudoku\_solver(6,4,7) simple\_sudoku\_solver(7,5,7) simple\_sudoku\_solver(1,4,8) simple\_sudoku\_solver(7,6,8) simple\_sudoku\_solver(8,4,9) simple\_sudoku\_solver(6,5,9) simple\_sudoku\_solver(2,6,9) simple\_sudoku\_solver(6,7,1) simple\_sudoku\_solver(3,9,1) simple\_sudoku\_solver(7,7,2) simple\_sudoku\_solver(3,8,2) simple\_sudoku\_solver(6,8,3) simple\_sudoku\_solver(1,9,3) simple\_sudoku\_solver(1,7,4) simple\_sudoku\_solver(5,8,4) simple\_sudoku\_solver(9,9,4) simple\_sudoku\_solver(9,8,5) simple\_sudoku\_solver(6,9,5) simple\_sudoku\_solver(2,7,6) simple\_sudoku\_solver(8,8,6) simple\_sudoku\_solver(4,9,6) simple\_sudoku\_solver(5,7,7) simple\_sudoku\_solver(2,8,7) simple\_sudoku\_solver(8,7,8) simple\_sudoku\_solver(4,8,8) simple\_sudoku\_solver(7,9,9)  SATISFIABLE  Models : 1  Calls : 1  Time : 0.039s (Solving: 0.02s 1st Model: 0.00s Unsat: 0.02s)  CPU Time : 0.000s |

Problem 4

|  |  |
| --- | --- |
| Input  Program | % First create a grid space which allows numbers from 1 to 16  % in a sudoku grid of size 16  { sudoku(Row, Col, N): Row=1..16, Col=1..16, Row>=Row1, Row<=Row1+3, Col>=Col1, Col<=Col1+3 }=1 :- N=1..16, Row1=4\*(0..3)+1, Col1=4\*(0..3)+1.  % Secondly, no 2 numbers are different in the same row and column  :- sudoku(Row, Col, N), sudoku(Row, Col, N1), N!=N1.  % Thirdly, no 2 columns are different given a row and number  :- sudoku(Row, Col, N), sudoku(Row, Col1, N), Col!=Col1.  % fourthly no 2 rows are different given a column and number  :- sudoku(Row, Col, N), sudoku(Row1, Col, N), Row!=Row1.  % Box 1  sudoku(1, 2, 6).  sudoku(2, 1, 15).  sudoku(2, 2, 11).  sudoku(3, 1, 13).  sudoku(3, 3, 9).  sudoku(3, 4, 12).  sudoku(4, 1, 2).  sudoku(4, 3, 16).  % Box 2  sudoku(5, 2, 15).  sudoku(5, 3, 11).  sudoku(5, 4, 10).  sudoku(6, 1, 12).  sudoku(6, 2, 13).  sudoku(7, 1, 5).  sudoku(7, 3, 6).  sudoku(7, 4, 1).  sudoku(8, 2, 2).  % Box 3  sudoku(9, 1, 10).  sudoku(9, 2, 7).  sudoku(9, 3, 15).  sudoku(9, 4, 11).  sudoku(10, 1, 9).  sudoku(11, 1, 1).  sudoku(11, 3, 4).  sudoku(11, 4, 6).  sudoku(12, 1, 16).  sudoku(12, 2, 14).  %Box 4  sudoku(13, 1, 11).  sudoku(13, 2, 10).  sudoku(13, 4, 15).  sudoku(14, 3, 12).  sudoku(15, 3, 5).  sudoku(16, 1, 3).  sudoku(16, 2, 16).  %Box 5  sudoku(1, 8, 8).  sudoku(2, 6, 16).  sudoku(2, 7, 14).  sudoku(4, 5, 11).  sudoku(4, 7, 15).  sudoku(4, 8, 10).  %Box 6  sudoku(5, 7, 16).  sudoku(5, 8, 2).  sudoku(6, 5, 4).  sudoku(6, 6, 1).  sudoku(6, 7, 5).  sudoku(6, 8, 6).  sudoku(7, 5, 12).  sudoku(7, 7, 9).  sudoku(8, 6, 10).  sudoku(8, 8, 11).  %Box 7  sudoku(9, 5, 16).  sudoku(10, 7, 1).  sudoku(11, 5, 9).  sudoku(11, 6, 13).  sudoku(12, 5, 7).  sudoku(12, 7, 10).  sudoku(12, 8, 15).  %Box 8  sudoku(13, 8, 16).  sudoku(14, 5, 1).  sudoku(14, 6, 4).  sudoku(14, 7, 6).  sudoku(15, 5, 8).  sudoku(15, 6, 12).  sudoku(15, 7, 13).  sudoku(16, 5, 10).  sudoku(16, 8, 7).  %Box 9  sudoku(1, 9, 11).  sudoku(1, 12, 15).  sudoku(2, 11, 12).  sudoku(3, 9, 3).  sudoku(3, 10, 16).  sudoku(3, 11, 14).  sudoku(4, 9, 1).  %Box 10  sudoku(5, 9, 13).  sudoku(5, 10, 8).  sudoku(5, 11, 9).  sudoku(5, 12, 12).  sudoku(6, 9, 2).  sudoku(6, 10, 3).  sudoku(7, 9, 15).  sudoku(7, 10, 11).  sudoku(7, 11, 10).  sudoku(7, 12, 7).  sudoku(8, 9, 6).  sudoku(8, 11, 5).  %Box 11  sudoku(9, 9, 12).  sudoku(9, 10, 13).  sudoku(10, 10, 2).  sudoku(10, 12, 16).  sudoku(11, 9, 7).  sudoku(11, 11, 11).  sudoku(12, 9, 4).  sudoku(12, 10, 6).  sudoku(12, 11, 1).  %Box 12  sudoku(13, 9, 9).  sudoku(13, 10, 12).  sudoku(13, 11, 13).  sudoku(14, 9, 16).  sudoku(15, 9, 10).  sudoku(15, 12, 11).  sudoku(16, 11, 6).  %Box 13  sudoku(1, 13, 14).  sudoku(1, 16, 16).  sudoku(2, 14, 6).  sudoku(3, 13, 15).  sudoku(3, 14, 11).  sudoku(3, 15, 10).  %Box 14  sudoku(6, 15, 11).  sudoku(6, 16, 10).  sudoku(7, 13, 16).  sudoku(7, 16, 3).  sudoku(8, 14, 13).  sudoku(8, 16, 9).  %Box 15  sudoku(9, 16, 6).  sudoku(10, 13, 10).  sudoku(10, 16, 11).  sudoku(11, 13, 3).  sudoku(11, 14, 16).  sudoku(12, 15, 13).  sudoku(12, 16, 8).  %Box 16  sudoku(13, 14, 1).  sudoku(13, 15, 5).  sudoku(13, 16, 4).  sudoku(14, 13, 11).  sudoku(14, 14, 10).  sudoku(15, 13, 2).  sudoku(15, 16, 14).  sudoku(16, 15, 12). |
| Command  Line | clingo .\P1\_Sudoku\_Solver\_Q4.lp 0 |
| Output  of clingo | clingo version 5.6.2  Reading from .\P1\_Sudoku\_Solver\_Q4.lp  Solving...  Answer: 1  sudoku(1,2,6) sudoku(2,1,15) sudoku(2,2,11) sudoku(3,1,13) sudoku(3,3,9) sudoku(3,4,12) sudoku(4,1,2) sudoku(4,3,16) sudoku(5,2,15) sudoku(5,3,11) sudoku(5,4,10) sudoku(6,1,12) sudoku(6,2,13) sudoku(7,1,5) sudoku(7,3,6) sudoku(7,4,1) sudoku(8,2,2) sudoku(9,1,10) sudoku(9,2,7) sudoku(9,3,15) sudoku(9,4,11) sudoku(10,1,9) sudoku(11,1,1) sudoku(11,3,4) sudoku(11,4,6) sudoku(12,1,16) sudoku(12,2,14) sudoku(13,1,11) sudoku(13,2,10) sudoku(13,4,15) sudoku(14,3,12) sudoku(15,3,5) sudoku(16,1,3) sudoku(16,2,16) sudoku(1,8,8) sudoku(2,6,16) sudoku(2,7,14) sudoku(4,5,11) sudoku(4,7,15) sudoku(4,8,10) sudoku(5,7,16) sudoku(5,8,2) sudoku(6,5,4) sudoku(6,6,1) sudoku(6,7,5) sudoku(6,8,6) sudoku(7,5,12) sudoku(7,7,9) sudoku(8,6,10) sudoku(8,8,11) sudoku(9,5,16) sudoku(10,7,1) sudoku(11,5,9) sudoku(11,6,13) sudoku(12,5,7) sudoku(12,7,10) sudoku(12,8,15) sudoku(13,8,16) sudoku(14,5,1) sudoku(14,6,4) sudoku(14,7,6) sudoku(15,5,8) sudoku(15,6,12) sudoku(15,7,13) sudoku(16,5,10) sudoku(16,8,7) sudoku(1,9,11) sudoku(1,12,15) sudoku(2,11,12) sudoku(3,9,3) sudoku(3,10,16) sudoku(3,11,14) sudoku(4,9,1) sudoku(5,9,13) sudoku(5,10,8) sudoku(5,11,9) sudoku(5,12,12) sudoku(6,9,2) sudoku(6,10,3) sudoku(7,9,15) sudoku(7,10,11) sudoku(7,11,10) sudoku(7,12,7) sudoku(8,9,6) sudoku(8,11,5) sudoku(9,9,12) sudoku(9,10,13) sudoku(10,10,2) sudoku(10,12,16) sudoku(11,9,7) sudoku(11,11,11) sudoku(12,9,4) sudoku(12,10,6) sudoku(12,11,1) sudoku(13,9,9) sudoku(13,10,12) sudoku(13,11,13) sudoku(14,9,16) sudoku(15,9,10) sudoku(15,12,11) sudoku(16,11,6) sudoku(1,13,14) sudoku(1,16,16) sudoku(2,14,6) sudoku(3,13,15) sudoku(3,14,11) sudoku(3,15,10) sudoku(6,15,11) sudoku(6,16,10) sudoku(7,13,16) sudoku(7,16,3) sudoku(8,14,13) sudoku(8,16,9) sudoku(9,16,6) sudoku(10,13,10) sudoku(10,16,11) sudoku(11,13,3) sudoku(11,14,16) sudoku(12,15,13) sudoku(12,16,8) sudoku(13,14,1) sudoku(13,15,5) sudoku(13,16,4) sudoku(14,13,11) sudoku(14,14,10) sudoku(15,13,2) sudoku(15,16,14) sudoku(16,15,12) sudoku(15,2,1) sudoku(1,3,1) sudoku(12,3,2) sudoku(16,4,2) sudoku(4,2,3) sudoku(8,3,3) sudoku(12,4,3) sudoku(1,1,4) sudoku(7,2,4) sudoku(15,4,4) sudoku(11,2,5) sudoku(1,4,5) sudoku(15,1,6) sudoku(5,1,7) sudoku(13,3,7) sudoku(2,4,7) sudoku(14,1,8) sudoku(3,2,8) sudoku(6,3,8) sudoku(10,4,8) sudoku(14,2,9) sudoku(6,4,9) sudoku(2,3,10) sudoku(10,2,12) sudoku(10,3,13) sudoku(14,4,13) sudoku(8,1,14) sudoku(16,3,14) sudoku(4,4,14) sudoku(8,4,16) sudoku(3,8,1) sudoku(2,5,2) sudoku(13,6,2) sudoku(9,7,2) sudoku(5,5,3) sudoku(9,6,3) sudoku(13,7,3) sudoku(2,8,3) sudoku(3,7,4) sudoku(10,8,4) sudoku(10,5,5) sudoku(3,6,5) sudoku(14,8,5) sudoku(3,5,6) sudoku(10,6,6) sudoku(4,6,7) sudoku(8,7,7) sudoku(7,6,8) sudoku(11,7,8) sudoku(1,6,9) sudoku(15,8,9) sudoku(12,6,11) sudoku(16,7,11) sudoku(1,7,12) sudoku(11,8,12) sudoku(1,5,13) sudoku(7,8,13) sudoku(13,5,14) sudoku(5,6,14) sudoku(9,8,14) sudoku(8,5,15) sudoku(16,6,15) sudoku(16,10,1) sudoku(8,12,1) sudoku(14,11,2) sudoku(3,12,2) sudoku(10,11,3) sudoku(14,12,3) sudoku(8,10,4) sudoku(4,11,4) sudoku(16,12,4) sudoku(16,9,5) sudoku(4,10,5) sudoku(9,12,5) sudoku(4,12,6) sudoku(15,10,7) sudoku(1,11,7) sudoku(2,9,8) sudoku(9,11,8) sudoku(13,12,8) sudoku(2,10,9) sudoku(12,12,9) sudoku(1,10,10) sudoku(11,12,10) sudoku(2,12,13) sudoku(10,9,14) sudoku(14,10,14) sudoku(6,12,14) sudoku(11,10,15) sudoku(15,11,15) sudoku(6,11,16) sudoku(2,13,1) sudoku(9,15,1) sudoku(5,16,1) sudoku(1,14,2) sudoku(7,15,2) sudoku(11,16,2) sudoku(15,14,3) sudoku(1,15,3) sudoku(5,13,4) sudoku(9,14,4) sudoku(2,15,4) sudoku(12,13,5) sudoku(5,14,5) sudoku(2,16,5) sudoku(13,13,6) sudoku(5,15,6) sudoku(6,13,7) sudoku(10,14,7) sudoku(14,15,7) sudoku(3,16,7) sudoku(16,13,8) sudoku(4,14,8) sudoku(8,15,8) sudoku(9,13,9) sudoku(16,14,9) sudoku(4,15,9) sudoku(8,13,12) sudoku(12,14,12) sudoku(4,16,12) sudoku(4,13,13) sudoku(16,16,13) sudoku(7,14,14) sudoku(11,15,14) sudoku(6,14,15) sudoku(10,15,15) sudoku(14,16,15) sudoku(15,15,16)  <Truncated>  sudoku(6,12,14) sudoku(11,10,15) sudoku(15,11,15) sudoku(6,11,16) sudoku(9,13,1) sudoku(2,15,1) sudoku(5,16,1) sudoku(1,14,2) sudoku(7,15,2) sudoku(11,16,2) sudoku(15,14,3) sudoku(1,15,3) sudoku(2,13,4) sudoku(5,14,4) sudoku(9,15,4) sudoku(5,13,5) sudoku(9,14,5) sudoku(2,16,5) sudoku(13,13,6) sudoku(5,15,6) sudoku(6,13,7) sudoku(10,14,7) sudoku(14,15,7) sudoku(3,16,7) sudoku(16,13,8) sudoku(4,14,8) sudoku(8,15,8) sudoku(12,13,9) sudoku(16,14,9) sudoku(4,15,9) sudoku(8,13,12) sudoku(12,14,12) sudoku(4,16,12) sudoku(4,13,13) sudoku(16,16,13) sudoku(7,14,14) sudoku(11,15,14) sudoku(6,14,15) sudoku(10,15,15) sudoku(14,16,15) sudoku(15,15,16)  SATISFIABLE  Models : 80  Calls : 1  Time : 1.093s (Solving: 1.00s 1st Model: 0.00s Unsat: 0.02s)  CPU Time : 0.063s |

Problem 5

|  |  |
| --- | --- |
| Input  Program | % First create a grid space which allows numbers from 1 to 9  % in a offset grid of size 9  1{ offset(X, Y, N): X=1..9, Y=1..9, X>=X1, X<=X1+2, Y>=Y1, Y<=Y1+2 }1 :- N=1..9, X1=3\*(0..2)+1, Y1=3\*(0..2)+1.  % Secondly, no 2 numbers are different in the same row and column  :- offset(X, Y, N), offset(X, Y, N1), N!=N1.  % Thirdly, no 2 columns are different given a row and number  :- offset(X, Y, N), offset(X, Y1, N), Y!=Y1.  % Fourthly no 2 rows are different given a column and number  :- offset(X, Y, N), offset(X1, Y, N), X!=X1.  % Fifthly check the same colored offsets do not have same number  :- offset(X, Y, N), offset(X1, Y1, N), X \ 3 == X1 \ 3, Y \ 3 == Y1 \ 3, 1{X != X1; Y != Y1}.  % Box 1  offset(1, 1, 4).  offset(2, 1, 6).  offset(2, 2, 1).  offset(2, 3, 9).  offset(3, 1, 5).  % Box 2  offset(4, 3, 4).  offset(5, 3, 5).  offset(6, 2, 6).  offset(6, 3, 3).  %Box 3  offset(7, 1, 7).  offset(7, 2, 8).  offset(7, 3, 1).  offset(8, 1, 9).  offset(9, 2, 5).  offset(9, 3, 2).  %Box 4  offset(1, 6, 6).  offset(2, 4, 3).  offset(2, 5, 2).  offset(2, 6, 7).  offset(3, 5, 4).  offset(3, 6, 1).  %Box 5  offset(4, 4, 6).  offset(4, 5, 9).  offset(4, 6, 2).  offset(5, 5, 3).  offset(5, 6, 4).  offset(6, 4, 1).  offset(6, 6, 5).  %Box 6  offset(7, 4, 2).  offset(7, 5, 6).  offset(8, 4, 5).  offset(8, 6, 3).  offset(9, 5, 1).  %Box 7  offset(1, 7, 1).  offset(1, 8, 3).  offset(2, 8, 5).  offset(3, 7, 9).  offset(3, 9, 6).  %Box 8  offset(4, 7, 3).  offset(4, 9, 5).  offset(5, 8, 6).  offset(6, 7, 7).  offset(6, 9, 4).  %Box 9  offset(7, 8, 4).  offset(7, 9, 3).  offset(8, 7, 2).  offset(8, 9, 1).  offset(9, 7, 6).  offset(9, 8, 7).  offset(9, 9, 9). |
| Command  Line | clingo .\P1\_Offset\_Sudoku\_Solver\_Q5.lp 0 |
| Output  of clingo | clingo version 5.6.2  Reading from .\P1\_Offset\_Sudoku\_Solver\_Q5.lp  Solving...  Answer: 1  offset(1,1,4) offset(2,1,6) offset(2,2,1) offset(2,3,9) offset(3,1,5) offset(4,3,4) offset(5,3,5) offset(6,2,6) offset(6,3,3) offset(7,1,7) offset(7,2,8) offset(7,3,1) offset(8,1,9) offset(9,2,5) offset(9,3,2) offset(1,6,6) offset(2,4,3) offset(2,5,2) offset(2,6,7) offset(3,5,4) offset(3,6,1) offset(4,4,6) offset(4,5,9) offset(4,6,2) offset(5,5,3) offset(5,6,4) offset(6,4,1) offset(6,6,5) offset(7,4,2) offset(7,5,6) offset(8,4,5) offset(8,6,3) offset(9,5,1) offset(1,7,1) offset(1,8,3) offset(2,8,5) offset(3,7,9) offset(3,9,6) offset(4,7,3) offset(4,9,5) offset(5,8,6) offset(6,7,7) offset(6,9,4) offset(7,8,4) offset(7,9,3) offset(8,7,2) offset(8,9,1) offset(9,7,6) offset(9,8,7) offset(9,9,9) offset(4,1,8) offset(7,7,5) offset(5,1,1) offset(5,4,7) offset(5,7,8) offset(5,2,9) offset(8,2,4) offset(8,5,7) offset(8,8,8) offset(5,9,2) offset(8,3,6) offset(6,1,2) offset(9,1,3) offset(9,4,4) offset(1,3,8) offset(1,9,7) offset(7,6,9) offset(2,9,8) offset(3,2,3) offset(3,8,2) offset(3,3,7) offset(9,6,8) offset(1,4,9) offset(1,2,2) offset(1,5,5) offset(4,2,7) offset(4,8,1) offset(2,7,4) offset(6,5,8) offset(6,8,9) offset(3,4,8)  SATISFIABLE  Models : 1  Calls : 1  Time : 0.046s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.01s)  CPU Time : 0.031s |

Problem 6

|  |  |
| --- | --- |
| Input  Program | % First create a grid space which allows numbers from 1 to 9  % in a antiknight grid of size 9  1{ antiknight(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.  % Secondly, no 2 numbers are different in the same row and column  :- antiknight(X, Y, N), antiknight(X, Y, N1), N!=N1.  % Thirdly, no 2 columns are different given a row and number  :- antiknight(X, Y, N), antiknight(X, Y1, N), Y!=Y1.  % Fourthly no 2 rows are different given a column and number  :- antiknight(X, Y, N), antiknight(X1, Y, N), X!=X1.  % Fifthly look for knight movements having same number and add as constraint  :- antiknight(X, Y, N), antiknight(X1, Y1, N), |X - X1| + |Y - Y1| == 3.  % BOX 1  antiknight(1, 2, 8).  antiknight(2, 1, 2).  % bOX 2  antiknight(4, 1, 6).  antiknight(6, 3, 8).  % box 3  antiknight(7, 1, 5).  % Box 4  antiknight(2, 4, 7).  antiknight(3, 6, 6).  % Box 5  antiknight(4, 4, 4).  antiknight(6, 6, 9).  % box 6  antiknight(7, 4, 8).  antiknight(8, 6, 5).  % box 7  antiknight(3, 9, 9).  % box 8  antiknight(4, 7, 7).  antiknight(6, 9, 5).  % box 9  antiknight(8, 9, 6).  antiknight(9, 8, 1). |
| Command  Line | clingo .\P1\_AntiKnight\_Sudoku\_Solver\_Q6.lp 0 |
| Output  of clingo | clingo version 5.6.2  Reading from .\P1\_AntiKnight\_Sudoku\_Solver\_Q6.lp  Solving...  Answer: 1  antiknight(1,2,8) antiknight(2,1,2) antiknight(4,1,6) antiknight(6,3,8) antiknight(7,1,5) antiknight(2,4,7) antiknight(3,6,6) antiknight(4,4,4) antiknight(6,6,9) antiknight(7,4,8) antiknight(8,6,5) antiknight(3,9,9) antiknight(4,7,7) antiknight(6,9,5) antiknight(8,9,6) antiknight(9,8,1) antiknight(2,3,1) antiknight(3,4,1) antiknight(6,1,1) antiknight(1,4,2) antiknight(5,2,2) antiknight(4,5,2) antiknight(1,1,3) antiknight(4,2,3) antiknight(2,6,3) antiknight(3,1,4) antiknight(1,6,4) antiknight(6,2,4) antiknight(3,2,5) antiknight(2,5,5) antiknight(4,3,5) antiknight(5,4,5) antiknight(1,3,6) antiknight(3,3,7) antiknight(5,1,7) antiknight(3,5,8) antiknight(1,5,9) antiknight(2,2,9) antiknight(5,3,9) antiknight(1,7,1) antiknight(5,6,1) antiknight(3,8,2) antiknight(3,7,3) antiknight(5,5,3) antiknight(2,9,4) antiknight(5,7,4) antiknight(1,8,5) antiknight(2,7,6) antiknight(6,4,6) antiknight(1,9,7) antiknight(6,5,7) antiknight(2,8,8) antiknight(4,6,8) antiknight(4,8,9) antiknight(4,9,1) antiknight(6,7,2) antiknight(6,8,3) antiknight(5,8,6) antiknight(5,9,8) antiknight(7,2,1) antiknight(8,3,2) antiknight(9,3,3) antiknight(8,4,3) antiknight(7,3,4) antiknight(9,2,6) antiknight(7,5,6) antiknight(8,2,7) antiknight(9,1,8) antiknight(8,1,9) antiknight(8,5,1) antiknight(7,6,2) antiknight(9,5,4) antiknight(9,6,7) antiknight(7,8,7) antiknight(8,7,8) antiknight(7,7,9) antiknight(9,4,9) antiknight(9,9,2) antiknight(7,9,3) antiknight(8,8,4) antiknight(9,7,5)  SATISFIABLE  Models : 1+  Calls : 1  Time : 0.031s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.016s |

Problem 7

|  |  |
| --- | --- |
| Input  Program | % Define the possible values for each cell  % As per the alphabet\_soln the following alphabets are present 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'Z'.  % 1 { alphabet\_soln(X,Y,V) : V={1;2;3;4;5;6;7;8;26} } 1 :- X=1..9, Y=1..9.  { alphabet\_soln(X, Y, N): X=1..9, Y=1..9, X>=X1, X<=X1+2, Y>=Y1, Y<=Y1+2 }=1 :- N=(1;2;3;4;5;6;7;8;26), X1=3\*(0..2)+1, Y1=3\*(0..2)+1.  % Secondly, no 2 numbers are different in the same row and column  :- alphabet\_soln(X, Y, N), alphabet\_soln(X, Y, N1), N!=N1.  % Each row must contain unique values  :- alphabet\_soln(X,Y,V), alphabet\_soln(X,Y1,V), Y!=Y1.  % Each column must contain unique values  :- alphabet\_soln(X,Y,V), alphabet\_soln(X1,Y,V), X!=X1.  % Define the grid.  % Box 1  alphabet\_soln(1, 1, 5).  alphabet\_soln(1, 3, 6).  alphabet\_soln(2, 1, 2).  alphabet\_soln(2, 3, 26).  alphabet\_soln(3, 2, 4).  % box 2  alphabet\_soln(6, 1, 7).  alphabet\_soln(6, 3, 5).  % Box 3  alphabet\_soln(7, 2, 7).  alphabet\_soln(8, 1, 26).  alphabet\_soln(8, 2, 6).  alphabet\_soln(9, 2, 5).  alphabet\_soln(9, 3, 1).  % Box 4  alphabet\_soln(1, 5, 2).  alphabet\_soln(2, 4, 5).  alphabet\_soln(2, 5, 6).  alphabet\_soln(3, 4, 8).  % Box 5  alphabet\_soln(4, 6, 5).  alphabet\_soln(5, 5, 8).  alphabet\_soln(5, 6, 26).  alphabet\_soln(6, 5, 3).  % Box 6  alphabet\_soln(7, 4, 2).  alphabet\_soln(7, 6, 6).  alphabet\_soln(9, 4, 26).  alphabet\_soln(9, 6, 8).  % Box 7  alphabet\_soln(1, 7, 4).  alphabet\_soln(1, 8, 26).  alphabet\_soln(3, 8, 2).  alphabet\_soln(3, 9, 6).  % Box 8  alphabet\_soln(4, 7, 26).  alphabet\_soln(4, 9, 4).  alphabet\_soln(5, 7, 2).  alphabet\_soln(5, 8, 3).  alphabet\_soln(6, 7, 6).  % Box 9  alphabet\_soln(7, 9, 26).  alphabet\_soln(8, 7, 8).  alphabet\_soln(8, 9, 7).  alphabet\_soln(9, 7, 3). |
| Command  Line | clingo .\P1\_Alphabet\_Sudoku\_Solver\_Q7.lp 0 |
| Output  of clingo | clingo version 5.6.2  Reading from .\P1\_Alphabet\_Sudoku\_Solver\_Q7.lp  Solving...  Answer: 1  alphabet\_soln(1,1,5) alphabet\_soln(1,3,6) alphabet\_soln(2,1,2) alphabet\_soln(2,3,26) alphabet\_soln(3,2,4) alphabet\_soln(6,1,7) alphabet\_soln(6,3,5) alphabet\_soln(7,2,7) alphabet\_soln(8,1,26) alphabet\_soln(8,2,6) alphabet\_soln(9,2,5) alphabet\_soln(9,3,1) alphabet\_soln(1,5,2) alphabet\_soln(2,4,5) alphabet\_soln(2,5,6) alphabet\_soln(3,4,8) alphabet\_soln(4,6,5) alphabet\_soln(5,5,8) alphabet\_soln(5,6,26) alphabet\_soln(6,5,3) alphabet\_soln(7,4,2) alphabet\_soln(7,6,6) alphabet\_soln(9,4,26) alphabet\_soln(9,6,8) alphabet\_soln(1,7,4) alphabet\_soln(1,8,26) alphabet\_soln(3,8,2) alphabet\_soln(3,9,6) alphabet\_soln(4,7,26) alphabet\_soln(4,9,4) alphabet\_soln(5,7,2) alphabet\_soln(5,8,3) alphabet\_soln(6,7,6) alphabet\_soln(7,9,26) alphabet\_soln(8,7,8) alphabet\_soln(8,9,7) alphabet\_soln(9,7,3) alphabet\_soln(3,1,1) alphabet\_soln(5,2,1) alphabet\_soln(1,4,1) alphabet\_soln(4,5,1) alphabet\_soln(8,6,1) alphabet\_soln(7,7,1) alphabet\_soln(2,8,1) alphabet\_soln(6,9,1) alphabet\_soln(4,2,2) alphabet\_soln(8,3,2) alphabet\_soln(6,6,2) alphabet\_soln(9,9,2) alphabet\_soln(7,1,3) alphabet\_soln(1,2,3) alphabet\_soln(4,3,3) alphabet\_soln(8,4,3) alphabet\_soln(3,6,3) alphabet\_soln(2,9,3) alphabet\_soln(9,1,4) alphabet\_soln(5,3,4) alphabet\_soln(6,4,4) alphabet\_soln(8,5,4) alphabet\_soln(2,6,4) alphabet\_soln(7,8,4) alphabet\_soln(7,5,5) alphabet\_soln(3,7,5) alphabet\_soln(8,8,5) alphabet\_soln(5,9,5) alphabet\_soln(5,1,6) alphabet\_soln(4,4,6) alphabet\_soln(9,8,6) alphabet\_soln(3,3,7) alphabet\_soln(5,4,7) alphabet\_soln(9,5,7) alphabet\_soln(1,6,7) alphabet\_soln(2,7,7) alphabet\_soln(4,8,7) alphabet\_soln(4,1,8) alphabet\_soln(2,2,8) alphabet\_soln(7,3,8) alphabet\_soln(6,8,8) alphabet\_soln(1,9,8) alphabet\_soln(6,2,26) alphabet\_soln(3,5,26)  <TRUNCATED>  Answer: 11  alphabet\_soln(1,1,5) alphabet\_soln(1,3,6) alphabet\_soln(2,1,2) alphabet\_soln(2,3,26) alphabet\_soln(3,2,4) alphabet\_soln(6,1,7) alphabet\_soln(6,3,5) alphabet\_soln(7,2,7) alphabet\_soln(8,1,26) alphabet\_soln(8,2,6) alphabet\_soln(9,2,5) alphabet\_soln(9,3,1) alphabet\_soln(1,5,2) alphabet\_soln(2,4,5) alphabet\_soln(2,5,6) alphabet\_soln(3,4,8) alphabet\_soln(4,6,5) alphabet\_soln(5,5,8) alphabet\_soln(5,6,26) alphabet\_soln(6,5,3) alphabet\_soln(7,4,2) alphabet\_soln(7,6,6) alphabet\_soln(9,4,26) alphabet\_soln(9,6,8) alphabet\_soln(1,7,4) alphabet\_soln(1,8,26) alphabet\_soln(3,8,2) alphabet\_soln(3,9,6) alphabet\_soln(4,7,26) alphabet\_soln(4,9,4) alphabet\_soln(5,7,2) alphabet\_soln(5,8,3) alphabet\_soln(6,7,6) alphabet\_soln(7,9,26) alphabet\_soln(8,7,8) alphabet\_soln(8,9,7) alphabet\_soln(9,7,3) alphabet\_soln(3,1,1) alphabet\_soln(5,2,1) alphabet\_soln(1,4,1) alphabet\_soln(4,5,1) alphabet\_soln(8,6,1) alphabet\_soln(7,7,1) alphabet\_soln(2,8,1) alphabet\_soln(6,9,1) alphabet\_soln(4,2,2) alphabet\_soln(8,3,2) alphabet\_soln(6,6,2) alphabet\_soln(9,9,2) alphabet\_soln(7,1,3) alphabet\_soln(2,2,3) alphabet\_soln(4,3,3) alphabet\_soln(8,4,3) alphabet\_soln(3,6,3) alphabet\_soln(1,9,3) alphabet\_soln(9,1,4) alphabet\_soln(5,3,4) alphabet\_soln(6,4,4) alphabet\_soln(7,5,4) alphabet\_soln(2,6,4) alphabet\_soln(8,8,4) alphabet\_soln(8,5,5) alphabet\_soln(3,7,5) alphabet\_soln(7,8,5) alphabet\_soln(5,9,5) alphabet\_soln(5,1,6) alphabet\_soln(4,4,6) alphabet\_soln(9,8,6) alphabet\_soln(3,3,7) alphabet\_soln(5,4,7) alphabet\_soln(9,5,7) alphabet\_soln(1,6,7) alphabet\_soln(2,7,7) alphabet\_soln(4,8,7) alphabet\_soln(4,1,8) alphabet\_soln(1,2,8) alphabet\_soln(7,3,8) alphabet\_soln(6,8,8) alphabet\_soln(2,9,8) alphabet\_soln(6,2,26) alphabet\_soln(3,5,26)  Answer: 12  alphabet\_soln(1,1,5) alphabet\_soln(1,3,6) alphabet\_soln(2,1,2) alphabet\_soln(2,3,26) alphabet\_soln(3,2,4) alphabet\_soln(6,1,7) alphabet\_soln(6,3,5) alphabet\_soln(7,2,7) alphabet\_soln(8,1,26) alphabet\_soln(8,2,6) alphabet\_soln(9,2,5) alphabet\_soln(9,3,1) alphabet\_soln(1,5,2) alphabet\_soln(2,4,5) alphabet\_soln(2,5,6) alphabet\_soln(3,4,8) alphabet\_soln(4,6,5) alphabet\_soln(5,5,8) alphabet\_soln(5,6,26) alphabet\_soln(6,5,3) alphabet\_soln(7,4,2) alphabet\_soln(7,6,6) alphabet\_soln(9,4,26) alphabet\_soln(9,6,8) alphabet\_soln(1,7,4) alphabet\_soln(1,8,26) alphabet\_soln(3,8,2) alphabet\_soln(3,9,6) alphabet\_soln(4,7,26) alphabet\_soln(4,9,4) alphabet\_soln(5,7,2) alphabet\_soln(5,8,3) alphabet\_soln(6,7,6) alphabet\_soln(7,9,26) alphabet\_soln(8,7,8) alphabet\_soln(8,9,7) alphabet\_soln(9,7,3) alphabet\_soln(3,1,1) alphabet\_soln(5,2,1) alphabet\_soln(1,4,1) alphabet\_soln(4,5,1) alphabet\_soln(8,6,1) alphabet\_soln(7,7,1) alphabet\_soln(6,8,1) alphabet\_soln(2,9,1) alphabet\_soln(4,2,2) alphabet\_soln(8,3,2) alphabet\_soln(6,6,2) alphabet\_soln(9,9,2) alphabet\_soln(7,1,3) alphabet\_soln(2,2,3) alphabet\_soln(4,3,3) alphabet\_soln(8,4,3) alphabet\_soln(3,6,3) alphabet\_soln(1,9,3) alphabet\_soln(9,1,4) alphabet\_soln(5,3,4) alphabet\_soln(6,4,4) alphabet\_soln(7,5,4) alphabet\_soln(2,6,4) alphabet\_soln(8,8,4) alphabet\_soln(8,5,5) alphabet\_soln(3,7,5) alphabet\_soln(7,8,5) alphabet\_soln(5,9,5) alphabet\_soln(5,1,6) alphabet\_soln(4,4,6) alphabet\_soln(9,8,6) alphabet\_soln(3,3,7) alphabet\_soln(5,4,7) alphabet\_soln(9,5,7) alphabet\_soln(1,6,7) alphabet\_soln(2,7,7) alphabet\_soln(4,8,7) alphabet\_soln(4,1,8) alphabet\_soln(1,2,8) alphabet\_soln(7,3,8) alphabet\_soln(2,8,8) alphabet\_soln(6,9,8) alphabet\_soln(6,2,26) alphabet\_soln(3,5,26)  SATISFIABLE  Models : 12  Calls : 1  Time : 0.062s (Solving: 0.05s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.000s |

Problem 8

|  |  |
| --- | --- |
| Input  Program | % 1. create a chess grid  {bishop(Row, 1..n)} :- Row=1..n.  % 2. Find bishops in the same diagonal and remove them.  :- bishop(Row1, Col1), bishop(Row2, Col2), |Row1 - Row2| == |Col1 - Col2|, Row1 != Row2.  % 3. Maximize the value of bishops in the solution  #maximize{ 1, Row, Col : bishop(Row, Col) }. |
| Command  Line | You should write multiple command lines below.   1. clingo -c n=3 .\P1\_Bishop\_Problem\_Q8.lp 0 2. clingo -c n=4 .\P1\_Bishop\_Problem\_Q8.lp 0 3. clingo -c n=5 .\P1\_Bishop\_Problem\_Q8.lp 0 4. clingo -c n=6 .\P1\_Bishop\_Problem\_Q8.lp 0 5. clingo -c n=7 .\P1\_Bishop\_Problem\_Q8.lp 0 6. clingo -c n=8 .\P1\_Bishop\_Problem\_Q8.lp 0 |
| Output  of clingo | **n=3**  clingo version 5.6.2  Reading from .\P1\_Bishop\_Problem\_Q8.lp  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 : 0.001s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.000s  **n=4**  clingo version 5.6.2  Reading from .\P1\_Bishop\_Problem\_Q8.lp  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 : 0.004s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.000s  **n=5**  clingo version 5.6.2  Reading from .\P1\_Bishop\_Problem\_Q8.lp  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 : 0.008s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.000s  **n=6**  clingo version 5.6.2  Reading from .\P1\_Bishop\_Problem\_Q8.lp  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 : 0.020s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.01s)  CPU Time : 0.016s  **n=7**  clingo version 5.6.2  Reading from .\P1\_Bishop\_Problem\_Q8.lp  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 : 0.485s (Solving: 0.49s 1st Model: 0.00s Unsat: 0.48s)  CPU Time : 0.359s  **n=8**  clingo version 5.6.2  Reading from .\P1\_Bishop\_Problem\_Q8.lp  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 : 16.070s (Solving: 16.06s 1st Model: 0.00s Unsat: 15.79s)  CPU Time : 13.547s |
| 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) = 2n-2** |

Problem 9

|  |  |
| --- | --- |
| Input  Program | {in(I, 1..k)}=1:- I=1..n.  :- in(I, S), in(J, S), in(I+J, S), I!=J. |
| Command  Line | You should write multiple command lines below.   1. clingo -c k=1 -c n=2 .\P1\_SchurrMod\_Q9.lp 0 2. clingo -c k=2 -c n=8 .\P1\_SchurrMod\_Q9.lp 0 3. clingo -c k=3 -c n=23 .\P1\_SchurrMod\_Q9.lp 0 4. clingo -c k=4 -c n=66 .\P1\_SchurrMod\_Q9.lp 0 |
| Output  of clingo | **k=1, n=2**  clingo version 5.6.2  Reading from .\P1\_SchurrMod\_Q9.lp  Solving...  Answer: 1  in(1,1) in(2,1)  SATISFIABLE  Models : 1  Calls : 1  Time : 0.000s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.000s  **k=2, n= 8**  clingo version 5.6.2  Reading from .\P1\_SchurrMod\_Q9.lp  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 : 0.000s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.000s  **k=3, n= 23**  clingo version 5.6.2  Reading from .\P1\_SchurrMod\_Q9.lp  Solving...  Answer: 1  in(1,2) in(2,2) in(3,3) in(4,2) in(5,3) in(6,3) in(7,3) in(8,2) in(9,1) in(10,1) in(11,2) in(12,1) in(13,1) in(14,1) in(15,1) in(16,2) in(17,1) in(18,1) in(19,3) in(20,1) in(21,3) in(22,2) in(23,3)  Answer: 2  in(1,2) in(2,2) in(3,3) in(4,2) in(5,3) in(6,3) in(7,3) in(8,2) in(9,1) in(10,1) in(11,2) in(12,1) in(13,1) in(14,1) in(15,1) in(16,1) in(17,1) in(18,1) in(19,3) in(20,1) in(21,3) in(22,2) in(23,3)  Answer: 3  in(1,2) in(2,2) in(3,3) in(4,2) in(5,3) in(6,3) in(7,3) in(8,2) in(9,1) in(10,1) in(11,2) in(12,1) in(13,1) in(14,1) in(15,1) in(16,1) in(17,2) in(18,1) in(19,3) in(20,1) in(21,3) in(22,2) in(23,3)  Answer: 4  in(3,2) in(1,3) in(2,3) in(4,3) in(5,2) in(6,2) in(7,2) in(8,3) in(9,1) in(10,1) in(11,3) in(12,1) in(13,1) in(14,1) in(15,1) in(16,1) in(17,1) in(18,1) in(19,2) in(20,1) in(21,2) in(22,3) in(23,2)  Answer: 5  in(3,2) in(1,3) in(2,3) in(4,3) in(5,2) in(6,2) in(7,2) in(8,3) in(9,1) in(10,1) in(11,3) in(12,1) in(13,1) in(14,1) in(15,1) in(16,1) in(17,3) in(18,1) in(19,2) in(20,1) in(21,2) in(22,3) in(23,2)  Answer: 6  in(3,2) in(1,3) in(2,3) in(4,3) in(5,2) in(6,2) in(7,2) in(8,3) in(9,1) in(10,1) in(11,3) in(12,1) in(13,1) in(14,1) in(15,1) in(16,3) in(17,1) in(18,1) in(19,2) in(20,1) in(21,2) in(22,3) in(23,2)  Answer: 7  in(1,1) in(2,1) in(3,3) in(4,1) in(5,3) in(6,3) in(7,3) in(8,1) in(9,2) in(10,2) in(11,1) in(12,2) in(13,2) in(14,2) in(15,2) in(16,1) in(17,2) in(18,2) in(19,3) in(20,2) in(21,3) in(22,1) in(23,3)  Answer: 8  in(1,1) in(2,1) in(3,3) in(4,1) in(5,3) in(6,3) in(7,3) in(8,1) in(9,2) in(10,2) in(11,1) in(12,2) in(13,2) in(14,2) in(15,2) in(16,2) in(17,1) in(18,2) in(19,3) in(20,2) in(21,3) in(22,1) in(23,3)  Answer: 9  in(1,1) in(2,1) in(3,3) in(4,1) in(5,3) in(6,3) in(7,3) in(8,1) in(9,2) in(10,2) in(11,1) in(12,2) in(13,2) in(14,2) in(15,2) in(16,2) in(17,2) in(18,2) in(19,3) in(20,2) in(21,3) in(22,1) in(23,3)  Answer: 10  in(1,1) in(2,1) in(3,2) in(4,1) in(5,2) in(6,2) in(7,2) in(8,1) in(9,3) in(10,3) in(11,1) in(12,3) in(13,3) in(14,3) in(15,3) in(16,1) in(17,3) in(18,3) in(19,2) in(20,3) in(21,2) in(22,1) in(23,2)  Answer: 11  in(1,1) in(2,1) in(3,2) in(4,1) in(5,2) in(6,2) in(7,2) in(8,1) in(9,3) in(10,3) in(11,1) in(12,3) in(13,3) in(14,3) in(15,3) in(16,3) in(17,1) in(18,3) in(19,2) in(20,3) in(21,2) in(22,1) in(23,2)  Answer: 12  in(1,1) in(2,1) in(3,2) in(4,1) in(5,2) in(6,2) in(7,2) in(8,1) in(9,3) in(10,3) in(11,1) in(12,3) in(13,3) in(14,3) in(15,3) in(16,3) in(17,3) in(18,3) in(19,2) in(20,3) in(21,2) in(22,1) in(23,2)  Answer: 13  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(17,3) in(18,3) in(19,1) in(20,3) in(21,1) in(22,2) in(23,1)  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 : 0.062s (Solving: 0.05s 1st Model: 0.00s Unsat: 0.00s)  CPU Time : 0.000s  **k=4, n= 66**  <Truncated>  in(3,3) in(1,4) in(2,4) in(4,4) in(5,3) in(6,3) in(7,3) in(8,4) in(9,1) in(10,1) in(11,4) in(12,1) in(13,1) in(14,1) in(15,1) in(16,4) in(17,1) in(18,1) in(19,3) in(20,1) in(21,3) in(22,4) in(23,3) in(24,2) in(25,4) in(26,2) in(27,2) in(28,2) in(29,2) in(30,2) in(31,2) in(32,4) in(33,2) in(34,2) in(35,3) in(36,2) in(37,3) in(38,2) in(39,4) in(40,2) in(41,2) in(42,2) in(43,2) in(44,2) in(45,2) in(46,4) in(47,2) in(48,2) in(49,2) in(50,3) in(51,3) in(52,3) in(53,4) in(54,1) in(55,1) in(56,1) in(57,1) in(58,4) in(59,1) in(60,1) in(61,1) in(62,1) in(63,4) in(64,3) in(65,3) in(66,3)  Answer: 9885  in(3,3) in(1,4) in(2,4) in(4,4) in(5,3) in(6,3) in(7,3) in(8,4) in(9,1) in(10,1) in(11,4) in(12,1) in(13,1) in(14,1) in(15,1) in(16,4) in(17,1) in(18,1) in(19,3) in(20,1) in(21,3) in(22,4) in(23,3) in(24,2) in(25,4) in(26,2) in(27,2) in(28,2) in(29,2) in(30,2) in(31,2) in(32,4) in(33,2) in(34,2) in(35,3) in(36,2) in(37,3) in(38,2) in(39,4) in(40,2) in(41,2) in(42,2) in(43,2) in(44,2) in(45,2) in(46,4) in(47,2) in(48,2) in(49,2) in(50,3) in(51,3) in(52,3) in(53,4) in(54,1) in(55,1) in(56,1) in(57,1) in(58,1) in(59,1) in(60,1) in(61,1) in(62,1) in(63,4) in(64,3) in(65,3) in(66,3)  Answer: 9886  in(3,3) in(1,4) in(2,4) in(4,4) in(5,3) in(6,3) in(7,3) in(8,4) in(9,1) in(10,1) in(11,4) in(12,1) in(13,1) in(14,1) in(15,1) in(16,4) in(17,1) in(18,1) in(19,3) in(20,1) in(21,3) in(22,4) in(23,3) in(24,2) in(25,4) in(26,2) in(27,2) in(28,2) in(29,2) in(30,2) in(31,2) in(32,4) in(33,2) in(34,2) in(35,3) in(36,2) in(37,3) in(38,2) in(39,4) in(40,2) in(41,2) in(42,2) in(43,2) in(44,2) in(45,2) in(46,4) in(47,2) in(48,2) in(49,2) in(50,3) in(51,3) in(52,3) in(53,4) in(54,1) in(55,1) in(56,1) in(57,1) in(58,1) in(59,1) in(60,1) in(61,1) in(62,1) in(63,4) in(64,3) in(65,3) in(66,4)  Answer: 9887  in(3,3) in(1,4) in(2,4) in(4,4) in(5,3) in(6,3) in(7,3) in(8,4) in(9,1) in(10,1) in(11,4) in(12,1) in(13,1) in(14,1) in(15,1) in(16,1) in(17,1) in(18,1) in(19,3) in(20,1) in(21,3) in(22,4) in(23,3) in(24,2) in(25,4) in(26,2) in(27,2) in(28,2) in(29,2) in(30,2) in(31,2) in(32,4) in(33,2) in(34,2) in(35,3) in(36,2) in(37,3) \*\*\* Info : (clingo): Queueing signal...  in(38,2) in(39,4) in(40,2) in(41,2) in(42,2) in(43,2) in(44,2) in(45,2) in(46,4) in(47,2) in(48,2) in(49,2) in(50,3) in(51,3) in(52,3) in(53,4) in(54,1) in(55,1) in(56,1) in(57,1) in(58,1) in(59,1) in(60,4) in(61,1) in(62,1) in(63,4) in(64,3) in(65,3) in(66,3)  \*\*\* Info : (clingo): INTERRUPTED by signal!  SATISFIABLE  INTERRUPTED : 1  Models : 9887+  Calls : 1  Time : 428.750s (Solving: 428.70s 1st Model: 394.19s Unsat: 0.00s)  CPU Time : 331.953s |
| Answer  to Questions | Fill in the values accordingly.   |  |  | | --- | --- | | Exact value of A(1) | 2 | | Exact value of A(2) | 8 | | Exact value of A(3) | 23 | | 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. | 66 | |