CSE 579
Module 5 Graded Assignment
Template for clingo Work

Problem 1

Input Program	Hint: you only need one program with a new term, whose value will be assigned to 3 or 4 in the command line. Program_1.txt %%%%%%%%%%%%%%%%%%%%%%%%%%%% % sort and object declaration %%%%%%%%%%%%%%%%%%%%%%%%%%%% % every block is a location location(B) :- block(B). % the table is a location location(table). %%%%%%%%%%%%%%%%%%%%%%%%%% % state description %%%%%%%%%%%%%%%%%%%%%%%%%% % two blocks can't be on the same block at the same time :- 2{on(BB,B,T)}, block(B), T = 0..m. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% % effect and preconditions of action %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% % effect of moving a block on(B,L,T+1) :- move(B,L,T). % concurrent actions are limited by num of grippers :- not {move(BB,LL,T)} grippers, T = 0..m-1. % a block can be moved only when it is clear :- move(B,L,T), on(B1,B,T). % a block can't be moved onto a block that is being moved also :- move(B,B1,T), move(B1,L,T).

	%%%%%%%%%%%%%%%%%%%%%%%%%% % domain independent axioms %%%%%%%%%%%%%%%%%%%%%%%%%% % fluents are initially exogenous 1{on(B,LL,0):location(LL)}1 :- block(B). % uniqueness and existence of value constraints :- not 1{on(B,LL,T)}1, block(B), T=1..m. % actions are exogenous {move(B,L,T)} :- block(B), location(L), T = 0..m-1. % commonsense law of inertia {on(B,L,T+1)} :- on(B,L,T), T < m. % limit blocks by max :- {on(B,table,A)} > max, A = 0..m. #show move/3. Blocks_Scenario_1.txt %%%%%%%%%%%%%%%%%% % File: Blocks_Scenario_1.txt %%%%%%%%%%%%%%%%%% block(1..6). :- not on(1,2,0; 2,table,0; 3,4,0; 4,table,0; 5,6,0; 6,table,0). :- not on(3,2,m; 2,1,m; 1,table,m; 6,5,m; 5,4,m; 4,table,m).
Command Lines	You should write multiple command lines below. Max = 3 clingo program_1.txt Blocks_Scenario_1.txt -c max=3 -c grippers=2 -c m=4 clingo program_1.txt Blocks_Scenario_1.txt -c max=3 -c grippers=2 -c m=5 Max =4 clingo program_1.txt Blocks_Scenario_1.txt -c max=4 -c grippers=2 -c m=3 clingo program_1.txt Blocks_Scenario_1.txt -c max=4 -c grippers=2 -c m=4

Outputs of clingo	You should write multiple outputs, one for each command. These outputs serve as the evidences of your answer to the following question.
	
	Hint 1: Let n be the maximal number of blocks that can be placed directly on the table. There should be 2 command lines and outputs for n=3, where
	• the 1st command line and output show k steps are not enough and
	• the 2nd command line and output show k+1 steps are enough.
	Similarly, there should be another 2 command lines and outputs for n=4.
	
	Hint 2: We do not give any limitation on the number of grippers.
	
	Max = 3
	
	i) Unsatisfiable
	`(py3.6) C:\Users\devra\Programming_Assignment_2>clingo program_1.txt Blocks_Scenario_1.txt -c max=3 -c grippers=2 -c m=4`
	`clingo version 5.4.0`
	`Reading from program_1.txt ...`
	`Solving...`
	`UNSATISFIABLE`
	
	`Models : 0`
	`Calls : 1`
	`Time : 0.016s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)`
	`CPU Time : 0.000s`
	ii) Satisfiable
	`(py3.6) C:\Users\devra\Programming_Assignment_2>clingo program_1.txt Blocks_Scenario_1.txt -c max=3 -c grippers=2 -c m=5`
	`clingo version 5.4.0`
	`Reading from program_1.txt ...`
	`Solving...`
	`Answer: 1`
	`move(3,6,0) move(5,4,0) move(1,5,1) move(1,table,2) move(2,5,2) move(2,1,3) move(3,5,3) move(3,2,4) move(6,5,4)`
	`SATISFIABLE`
	
	`Models : 1+`
	`Calls : 1`
	`Time : 0.016s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)`
	`CPU Time : 0.000s`
	
	Max = 4
	i) Unsatisfiable

```
(py3.6) C:\Users\devra\Programming_Assignment_2>clingo program_1.txt Blocks_
Scenario_1.txt -c max=4 -c grippers=2 -c m=3
clingo version 5.4.0
Reading from program_1.txt ...
Solving...
UNSATISFIABLE

Models       : 0
Calls        : 1
Time         : 0.000s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s
```

ii) Satisfiable

```
(py3.6) C:\Users\devra\Programming_Assignment_2>clingo program_1.txt Blocks_
Scenario_1.txt -c max=4 -c grippers=2 -c m=4
clingo version 5.4.0
Reading from program_1.txt ...
Solving...
Answer: 1
move(1,table,0) move(2,4,1) move(3,table,1) move(2,1,2) move(5,4,2) move(3,2
,3) move(6,5,3)
SATISFIABLE

Models       : 1+
Calls        : 1
Time         : 0.017s (Solving: 0.02s 1st Model: 0.02s Unsat: 0.00s)
CPU Time     : 0.000s
```

Answer to Questions

Fill in the following table that lists the minimum number of steps to solve the modified block world problem for different values of n, where n is the maximal number of blocks that can be placed directly on the table.

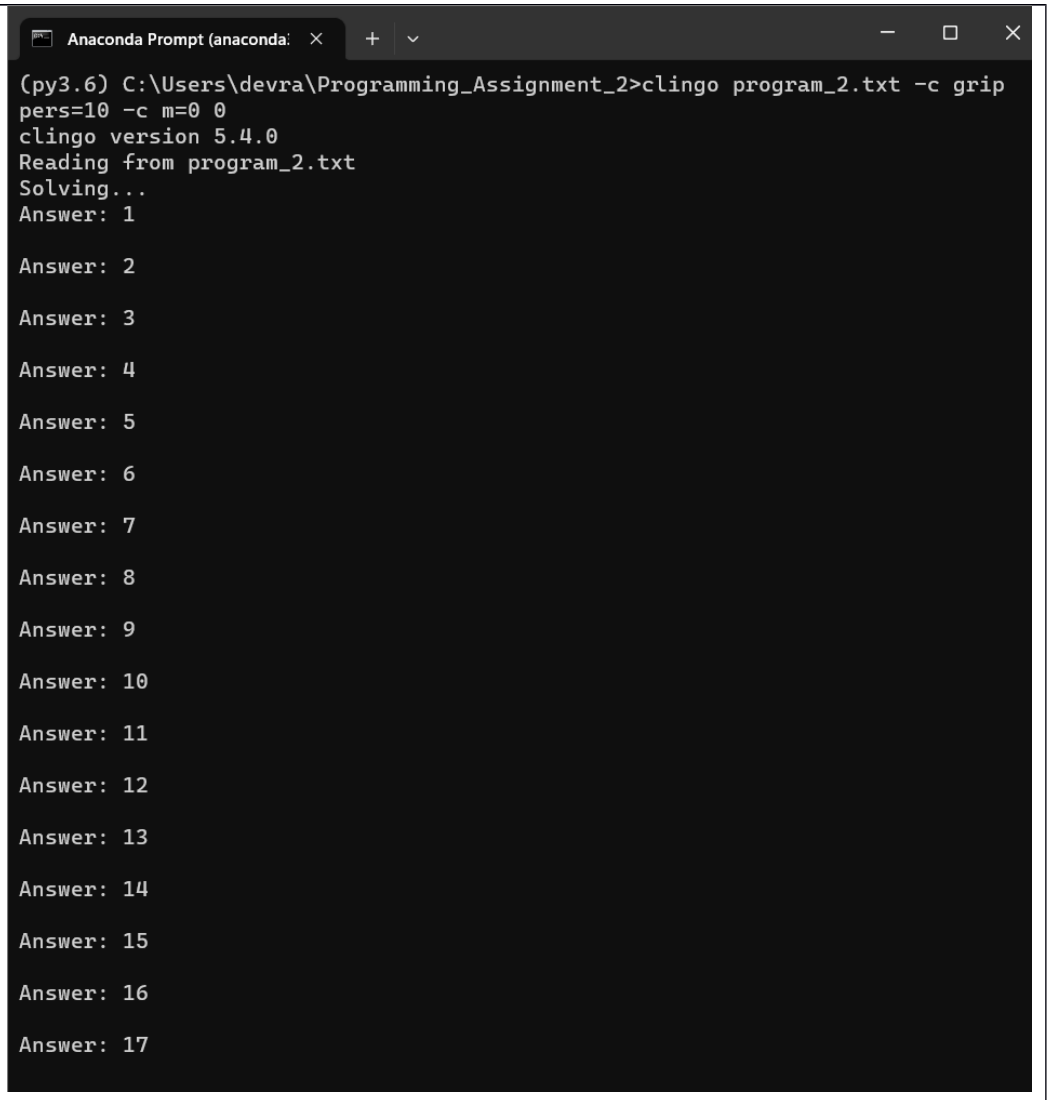
n	Number of steps
3	5
4	4

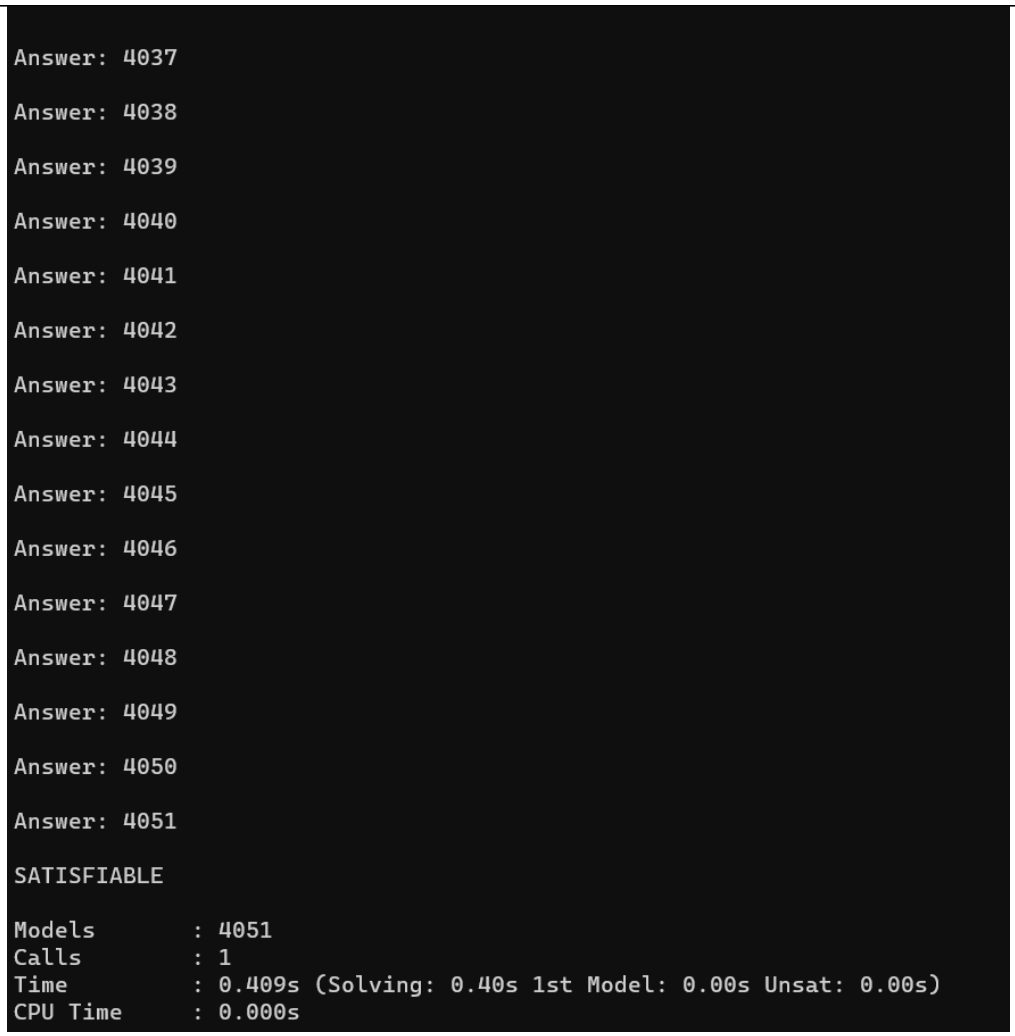
Problem 2

Input Program	Hint 1: You don't need to represent any scenario since you want to find out all possible valid states with 6 blocks. Think about the value of m. Hint 2: You don't need to consider the limitation of the maximum number of blocks on the table. That's only required in Problem 1. Program_2.txt %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
% sort and object declaration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% every block is a location
location(B) :- block(B).

% the table is a location
location(table).

%%%%%%%%%%%%%%%%%%%%%%%%%%
% state description
%%%%%%%%%%%%%%%%%%%%%%%%%%

% two blocks can't be on the same block at the same time
:- 2{on(BB,B,T)}, block(B), T = 0..m.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% effect and preconditions of action
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% effect of moving a block
on(B,L,T+1) :- move(B,L,T).

% concurrent actions are limited by num of grippers
:- not {move(BB,LL,T)} grippers, T = 0..m-1.

% a block can be moved only when it is clear
:- move(B,L,T), on(B1,B,T).

% a block can't be moved onto a block that is being moved also
:- move(B,B1,T), move(B1,L,T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% domain independent axioms
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% fluents are initially exogenous
1{on(B,LL,0):location(LL)}1 :- block(B).

% uniqueness and existence of value constraints
:- not 1{on(B,LL,T)}1, block(B), T=1..m.

% actions are exogenous
{move(B,L,T)} :- block(B), location(L), T = 0..m-1.
```

	% commonsense law of inertia {on(B,L,T+1)} :- on(B,L,T), T < m. block(1..6). one_on_one(B1,B2,T) :- on(B2,B1,T). one_on_one(B1,B3,T) :- one_on_one(B1,B2,T), one_on_one(B2,B3,T). :- on(B1,B2,T), one_on_one(B1,B2,T). :- on(B1,B1,T). #show move/3.
Command Line	clingo program_2.txt -c grippers=10 -c m=0 0

Output of clingo	

	 Complete output in Program_2_Output.docx file attached to this submission.
Answer to Questions	How many valid states are there when there are 6 blocks? (Note that the limitation of blocks introduced in question 1 is not considered here.) There are **4051** valid states.

Problem 3

Reading: A plan may allow multiple actions happening at the same time, e.g., when we have multiple robots working together to increase efficiency. However, if there is a little bit delay on one action, then we may get unexpected results. For example, when 2 robots are moving 2 adjacent blocks to the left at the same time, if there is a delay for the robot on the left-hand side, then these 2 robots may hit with each other. To make sure that our plan will get the expected result, we introduce the restriction "serializable" on the actions happening at the same time. This restriction simply says that, even if some actions in the same time stamp happen in serial with arbitrary ordering, the final result would be the same.

Input Program	Hint: the number of grippers is unlimited, meaning that you can have as many movements as you want as far as the movements are serializable. Program_3.txt %%%%%%%%%%%%%%%%%%%%%%%%%%%% % sort and object declaration %%%%%%%%%%%%%%%%%%%%%%%%%%%% % every block is a location location(B) :- block(B). % the table is a location location(table). %%%%%%%%%%%%%%%%%%%%%%%%%%%% % state description %%%%%%%%%%%%%%%%%%%%%%%%%%%% % two blocks can't be on the same block at the same time :- 2{on(BB,B,T)}, block(B), T = 0..m. %% % effect and preconditions of action %% % effect of moving a block on(B,L,T+1) :- move(B,L,T). % concurrent actions are limited by num of grippers :- not {move(BB,LL,T)} grippers, T = 0..m-1.

```
% a block can be moved only when it is clear
:- move(B,L,T), on(B1,B,T).

% a block can't be moved onto a block that is being moved also
:- move(B,B1,T), move(B1,L,T).

%%%%%%%%%%%%%%%%%%%%%%%%%%
% domain independent axioms
%%%%%%%%%%%%%%%%%%%%%%%%%%

% fluents are initially exogenous
1{on(B,LL,0):location(LL)}1 :- block(B).

% uniqueness and existence of value constraints
:- not 1{on(B,LL,T)}1, block(B), T=1..m.

% actions are exogenous
{move(B,L,T)} :- block(B), location(L), T = 0..m-1.

% commonsense law of inertia
{on(B,L,T+1)} :- on(B,L,T), T < m.

% action serialization
:- move(X1,Y1,A), on(X2,Y1,A), move(X2,Y2,A), block(X2), A = 1..m-1.

#show move/3.
```

Blocks_Scenario_2.txt

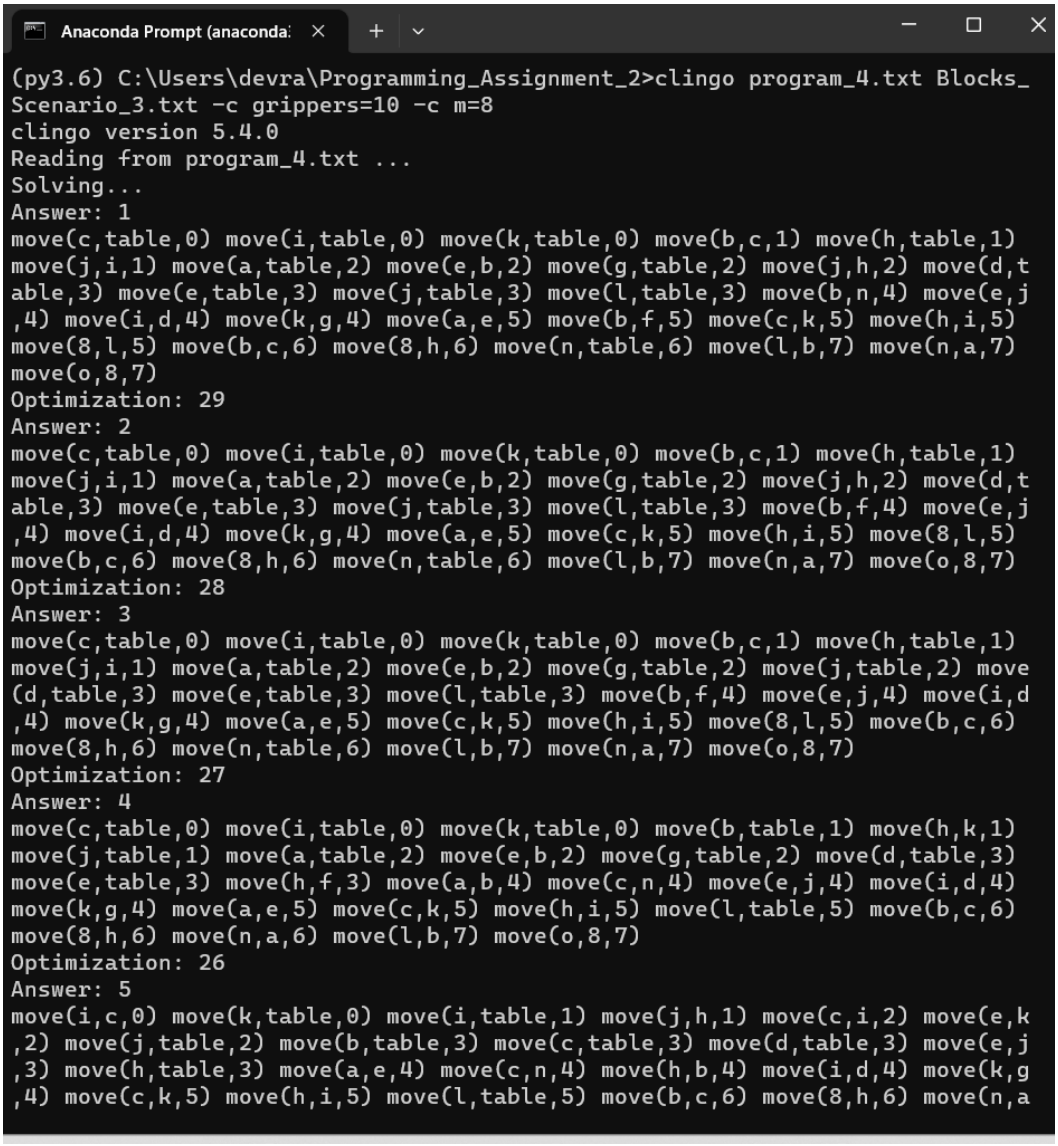
```
%%%%%%%%%%%%%%%%%%
% File: Blocks_Scenario_2.txt
%%%%%%%%%%%%%%%%%%

block(a;b;c;d;e;f;g;h;i;j;k;l;m;n;o).

% initial state
:- not on(m,table,0; l,m,0; a,l,0; b,a,0; c,b,0; o,table,0; n,o,0; d,n,0; e,d,0;
j,e,0; k,j,0; f,table,0; g,f,0; h,g,0; i,h,0).

% maxstep
:- not on(e,j,m; a,e,m; n,a,m; i,d,m; h,i,m; m,h,m; o,m,m; k,g,m; c,k,m;
b,c,m; l,b,m).
```

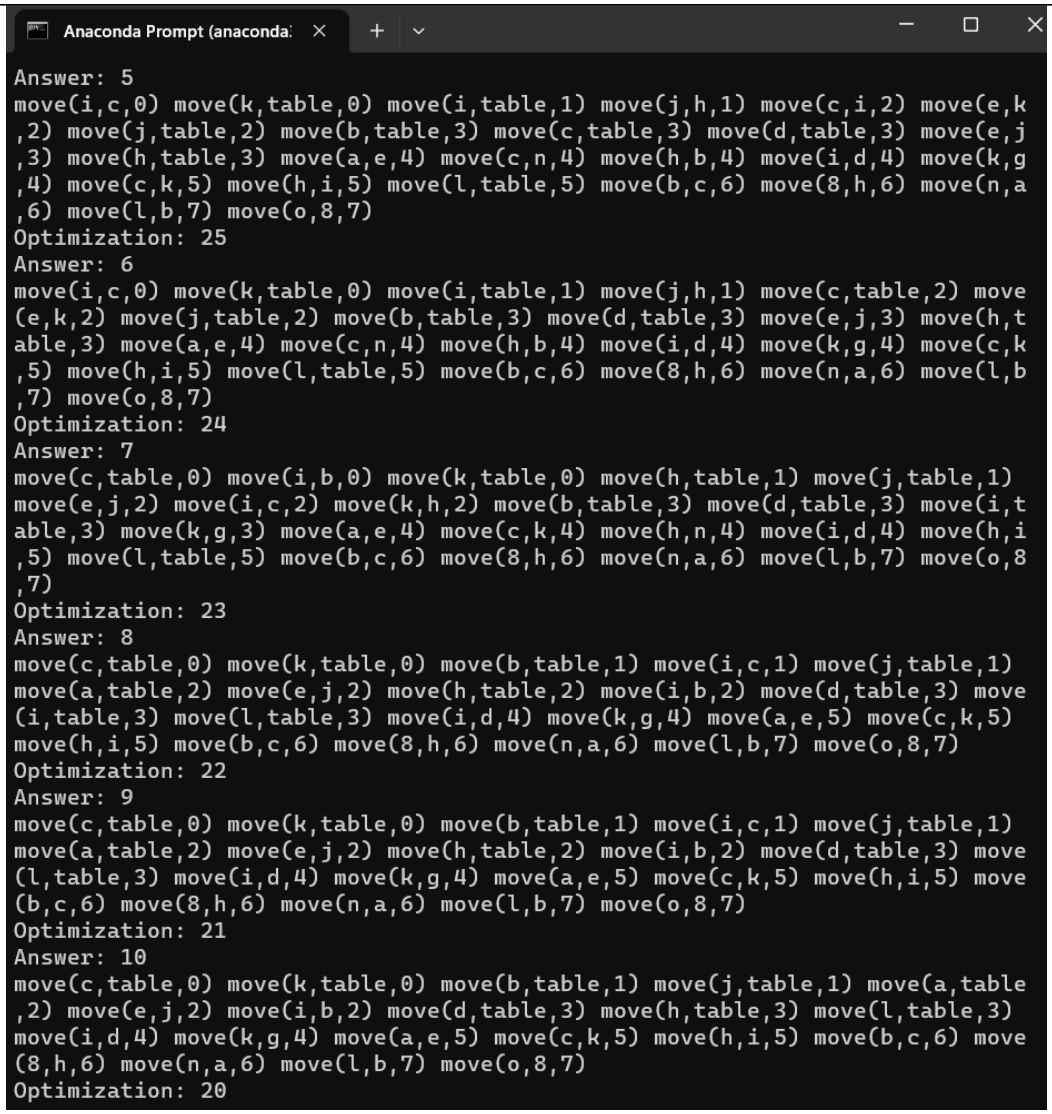
Command Line	Please only show the command line that outputs the minimal length plan. clingo program_3.txt Blocks_Scenario_2.txt -c grippers=10 -c m=8
Output of clingo	(py3.6) C:\Users\devra\Programming_Assignment_2>clingo program_3.txt Blocks_Scenario_2.txt -c grippers=10 -c m=8 clingo version 5.4.0 Reading from program_3.txt ... Solving... Answer: 1 move(c,table,0) move(i,table,0) move(k,table,0) move(b,table,1) move(h,c,1) move(j,table,1) move(e,h,2) move(i,b,2) move(k,g,2) move(a,k,3) move(d,table,3) move(e,j,3) move(a,e,4) move(h,table,4) move(i,d,4) move(l,table,4) move(n,table,4) move(b,a,5) move(c,k,5) move(h,i,5) move(l,n,5) move(8,o,5) move(b,c,6) move(l,table,6) move(8,h,6) move(l,b,7) move(n,a,7) move(o,8,7) SATISFIABLE Models : 1+ Calls : 1 Time : 0.578s (Solving: 0.03s 1st Model: 0.03s Unsat: 0.00s) CPU Time : 0.516s

Problem 4

Input Program	Program_4.txt %%%%%%%%%%%%%%%%%%%%%%%%%%%% % sort and object declaration %%%%%%%%%%%%%%%%%%%%%%%%%%%% % every block is a location location(B) :- block(B). % the table is a location location(table). %%%%%%%%%%%%%%%%%%%%%%%%%% % state description %%%%%%%%%%%%%%%%%%%%%%%%%% % two blocks can't be on the same block at the same time :- 2{on(BB,B,T)}, block(B), T = 0..m. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% % effect and preconditions of action %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% % effect of moving a block on(B,L,T+1) :- move(B,L,T). % concurrent actions are limited by num of grippers :- not {move(BB,LL,T)} grippers, T = 0..m-1. % a block can be moved only when it is clear :- move(B,L,T), on(B1,B,T). % a block can't be moved onto a block that is being moved also :- move(B,B1,T), move(B1,L,T). %%%%%%%%%%%%%%%%%%%%%%%%%% % domain independent axioms %%%%%%%%%%%%%%%%%%%%%%%%%% % fluents are initially exogenous 1{on(B,LL,0):location(LL)}1 :- block(B).

	% uniqueness and existence of value constraints :- not 1{on(B,LL,T)}1, block(B), T=1..m. % actions are exogenous {move(B,L,T)} :- block(B), location(L), T = 0..m-1. % commonsense law of inertia {on(B,L,T+1)} :- on(B,L,T), T < m. % action serialization :- move(X1,Y1,A), on(X2,Y1,A), move(X2,Y2,A), block(X2), A = 1..m-1. % moves minimization #minimize{1,B,L,T:move(B,L,T)}. #show move/3. Blocks_Scenario_3.txt %%%%%%%%%%%%%%%%% % File: blocks-scenario.txt %%%%%%%%%%%%%%%%% block(a;b;c;d;e;f;g;h;i;j;k;l;m;n;o). % initial state :- not on(m,table,0; l,m,0; a,l,0; b,a,0; c,b,0; o,table,0; n,o,0; d,n,0; e,d,0; j,e,0; k,j,0; f,table,0; g,f,0; h,g,0; i,h,0). % maxstep :- not on(e,j,m; a,e,m; n,a,m; i,d,m; h,i,m; m,h,m; o,m,m; k,g,m; c,k,m; b,c,m; l,b,m).
Command Line	You should write multiple command lines below. M = 8 clingo program_4.txt Blocks_Scenario_3.txt -c grippers=10 -c m=8 M = 9 clingo program_4.txt Blocks_Scenario_3.txt -c grippers=10 -c m=9 M = 10 clingo program_4.txt Blocks_Scenario_3.txt -c grippers=10 -c m=10 -t4
Output	You should write multiple outputs, one for each command. These outputs serve

of clingo	as the evidences of your answer to the question below. M = 8 


Anaconda Prompt (anaconda:
Answer: 5
move(i,c,0) move(k,table,0) move(i,table,1) move(j,h,1) move(c,i,2) move(e,k
,2) move(j,table,2) move(b,table,3) move(c,table,3) move(d,table,3) move(e,j
,3) move(h,table,3) move(a,e,4) move(c,n,4) move(h,b,4) move(i,d,4) move(k,g
,4) move(c,k,5) move(h,i,5) move(l,table,5) move(b,c,6) move(8,h,6) move(n,a
,6) move(l,b,7) move(o,8,7)
Optimization: 25
Answer: 6
move(i,c,0) move(k,table,0) move(i,table,1) move(j,h,1) move(c,table,2) move
(e,k,2) move(j,table,2) move(b,table,3) move(d,table,3) move(e,j,3) move(h,t
able,3) move(a,e,4) move(c,n,4) move(h,b,4) move(i,d,4) move(k,g,4) move(c,k
,5) move(h,i,5) move(l,table,5) move(b,c,6) move(8,h,6) move(n,a,6) move(l,b
,7) move(o,8,7)
Optimization: 24
Answer: 7
move(c,table,0) move(i,b,0) move(k,table,0) move(h,table,1) move(j,table,1)
move(e,j,2) move(i,c,2) move(k,h,2) move(b,table,3) move(d,table,3) move(i,t
able,3) move(k,g,3) move(a,e,4) move(c,k,4) move(h,n,4) move(i,d,4) move(h,i
,5) move(l,table,5) move(b,c,6) move(8,h,6) move(n,a,6) move(l,b,7) move(o,8
,7)
Optimization: 23
Answer: 8
move(c,table,0) move(k,table,0) move(b,table,1) move(i,c,1) move(j,table,1)
move(a,table,2) move(e,j,2) move(h,table,2) move(i,b,2) move(d,table,3) move
(i,table,3) move(l,table,3) move(i,d,4) move(k,g,4) move(a,e,5) move(c,k,5)
move(h,i,5) move(b,c,6) move(8,h,6) move(n,a,6) move(l,b,7) move(o,8,7)
Optimization: 22
Answer: 9
move(c,table,0) move(k,table,0) move(b,table,1) move(i,c,1) move(j,table,1)
move(a,table,2) move(e,j,2) move(h,table,2) move(i,b,2) move(d,table,3) move
(l,table,3) move(i,d,4) move(k,g,4) move(a,e,5) move(c,k,5) move(h,i,5) move
(b,c,6) move(8,h,6) move(n,a,6) move(l,b,7) move(o,8,7)
Optimization: 21
Answer: 10
move(c,table,0) move(k,table,0) move(b,table,1) move(j,table,1) move(a,table
,2) move(e,j,2) move(i,b,2) move(d,table,3) move(h,table,3) move(l,table,3)
move(i,d,4) move(k,g,4) move(a,e,5) move(c,k,5) move(h,i,5) move(b,c,6) move
(8,h,6) move(n,a,6) move(l,b,7) move(o,8,7)
Optimization: 20

```
move(c,table,0) move(k,table,0) move(b,table,1) move(i,c,1) move(j,table,1)
move(a,table,2) move(e,j,2) move(h,table,2) move(i,b,2) move(d,table,3) move
(i,table,3) move(l,table,3) move(i,d,4) move(k,g,4) move(a,e,5) move(c,k,5)
move(h,i,5) move(b,c,6) move(8,h,6) move(n,a,6) move(l,b,7) move(o,8,7)
Optimization: 22
Answer: 9
move(c,table,0) move(k,table,0) move(b,table,1) move(i,c,1) move(j,table,1)
move(a,table,2) move(e,j,2) move(h,table,2) move(i,b,2) move(d,table,3) move
(l,table,3) move(i,d,4) move(k,g,4) move(a,e,5) move(c,k,5) move(h,i,5) move
(b,c,6) move(8,h,6) move(n,a,6) move(l,b,7) move(o,8,7)
Optimization: 21
Answer: 10
move(c,table,0) move(k,table,0) move(b,table,1) move(j,table,1) move(a,table
,2) move(e,j,2) move(i,b,2) move(d,table,3) move(h,table,3) move(l,table,3)
move(i,d,4) move(k,g,4) move(a,e,5) move(c,k,5) move(h,i,5) move(b,c,6) move
(8,h,6) move(n,a,6) move(l,b,7) move(o,8,7)
Optimization: 20
Answer: 11
move(i,table,0) move(k,c,0) move(h,table,1) move(j,table,1) move(k,table,1)
move(c,h,2) move(e,j,2) move(k,g,2) move(b,table,3) move(c,k,3) move(d,table
,3) move(a,e,4) move(b,c,4) move(i,d,4) move(h,i,5) move(l,b,5) move(n,a,5)
move(8,h,6) move(o,8,7)
Optimization: 19
Answer: 12
move(i,table,0) move(k,table,0) move(h,table,1) move(j,table,1) move(c,h,2)
move(e,j,2) move(k,g,2) move(b,table,3) move(c,k,3) move(d,table,3) move(a,e
,4) move(b,c,4) move(i,d,4) move(h,i,5) move(l,b,5) move(n,a,5) move(8,h,6)
move(o,8,7)
Optimization: 18
OPTIMUM FOUND

Models       : 12
  Optimum    : yes
Optimization : 18
Calls        : 1
Time         : 1.163s (Solving: 0.61s 1st Model: 0.01s Unsat: 0.36s)
CPU Time     : 1.125s
```

M = 9

```
(py3.6) C:\Users\devra\Programming_Assignment_2>clingo program_4.txt Blocks_
Scenario_3.txt -c grippers=10 -c m=9
clingo version 5.4.0
Reading from program_4.txt ...
Solving...
Answer: 1
move(c,table,0) move(i,table,0) move(k,table,0) move(b,table,1) move(j,table
,1) move(a,table,2) move(e,table,2) move(h,table,2) move(d,table,3) move(g,t
able,3) move(l,table,3) move(i,d,4) move(k,g,4) move(c,k,5) move(e,j,5) move
(h,i,5) move(n,l,5) move(a,e,6) move(b,c,6) move(9,h,6) move(n,f,6) move(l,b
,7) move(n,a,7) move(o,9,7)
Optimization: 24
Answer: 2
move(c,table,0) move(i,table,0) move(k,table,0) move(b,table,1) move(h,table
,1) move(j,table,1) move(a,table,2) move(e,table,2) move(d,table,3) move(g,t
able,3) move(l,table,3) move(i,d,4) move(k,g,4) move(c,k,5) move(e,j,5) move
(h,i,5) move(a,e,6) move(b,c,6) move(9,h,6) move(n,f,6) move(l,b,7) move(n,a
,7) move(o,9,7)
Optimization: 23
Answer: 3
move(c,table,0) move(k,table,0) move(b,table,1) move(i,table,1) move(j,table
,1) move(a,table,2) move(e,j,2) move(h,table,2) move(d,table,3) move(g,table
,3) move(l,table,3) move(i,d,4) move(k,g,4) move(c,k,5) move(h,i,5) move(a,e
,6) move(b,c,6) move(9,h,6) move(n,f,6) move(l,b,7) move(n,a,7) move(o,9,7)
Optimization: 22
Answer: 4
move(c,table,0) move(k,table,0) move(b,table,1) move(i,c,1) move(j,table,1)
move(a,table,2) move(e,table,2) move(h,table,2) move(d,table,3) move(l,table
,3) move(i,d,4) move(k,g,4) move(c,k,5) move(e,j,5) move(h,i,5) move(a,e,6)
move(b,c,6) move(9,h,6) move(l,b,7) move(n,a,7) move(o,9,8)
Optimization: 21
Answer: 5
move(c,table,0) move(k,table,0) move(b,table,1) move(i,c,1) move(j,table,1)
move(a,table,2) move(e,j,2) move(h,table,2) move(d,table,3) move(l,table,3)
move(i,d,4) move(k,g,4) move(c,k,5) move(h,i,5) move(a,e,6) move(b,c,6) move
(9,h,6) move(l,b,7) move(n,a,7) move(o,9,8)
Optimization: 20
Answer: 6
move(i,table,0) move(k,c,0) move(h,i,1) move(j,table,1) move(e,table,2) move
```

```
Answer: 6
move(i,table,0) move(k,c,0) move(h,i,1) move(j,table,1) move(e,table,2) move
(k,g,2) move(c,k,3) move(d,table,3) move(h,table,3) move(b,c,4) move(i,d,4)
move(a,n,5) move(e,j,5) move(h,i,5) move(a,e,6) move(l,b,6) move(9,h,7) move
(n,a,7) move(o,9,8)
Optimization: 19
Answer: 7
move(i,table,0) move(k,c,0) move(h,table,1) move(j,table,1) move(e,table,2)
move(k,g,2) move(c,k,3) move(d,table,3) move(b,c,4) move(i,d,4) move(a,n,5)
move(e,j,5) move(h,i,5) move(a,e,6) move(l,b,6) move(9,h,7) move(n,a,7) move
(o,9,8)
Optimization: 18
Answer: 8
move(i,table,0) move(k,c,0) move(h,table,1) move(j,table,1) move(e,table,2)
move(k,g,2) move(c,k,3) move(d,table,3) move(b,c,4) move(e,j,4) move(i,d,4)
move(a,e,5) move(h,i,5) move(l,b,6) move(9,h,7) move(n,a,7) move(o,9,8)
Optimization: 17
Answer: 9
move(i,table,0) move(k,c,0) move(h,table,1) move(j,table,1) move(e,j,2) move
(k,g,2) move(c,k,3) move(d,table,3) move(b,c,4) move(i,d,4) move(a,e,5) move
(h,i,5) move(l,b,6) move(9,h,7) move(n,a,7) move(o,9,8)
Optimization: 16
OPTIMUM FOUND

Models       : 9
  Optimum    : yes
Optimization : 16
Calls        : 1
Time         : 32.754s (Solving: 31.95s 1st Model: 0.00s Unsat: 7.05s)
CPU Time     : 31.969s
```

M = 10

```
(py3.6) C:\Users\devra\Programming_Assignment_2>clingo program_4.txt Blocks_
Scenario_3.txt -c grippers=10 -c m=10 -t4
clingo version 5.4.0
Reading from program_4.txt ...
Solving...
Progression : [1;inf]
Progression : [2;inf]
Answer: 1
move(c,table,0) move(i,table,0) move(k,table,0) move(b,table,1) move(j,table
,1) move(a,table,2) move(e,table,2) move(h,table,2) move(d,table,3) move(g,t
able,3) move(l,table,4) move(n,table,4) move(i,d,5) move(k,g,5) move(c,k,6)
move(e,j,6) move(h,i,6) move(a,e,7) move(b,c,7) move(10,h,7) move(l,b,8) mov
e(o,10,8) move(n,a,9)
Optimization: 23
Progression : [ 3;23] (Error: 6.66667)
Progression : [ 4;23] (Error: 4.75)
Progression : [ 5;23] (Error: 3.6)
Progression : [ 6;23] (Error: 2.83333)
Progression : [ 7;23] (Error: 2.28571)
Answer: 2
move(c,table,0) move(i,table,0) move(k,table,0) move(b,table,1) move(h,table
,1) move(j,table,1) move(a,table,2) move(e,table,2) move(d,table,3) move(l,t
able,4) move(n,table,4) move(i,d,5) move(k,g,5) move(c,k,6) move(e,j,6) move
(h,i,6) move(a,e,7) move(b,c,7) move(10,h,7) move(l,b,8) move(o,10,8) move(n
,a,9)
Optimization: 22
Progression : [ 8;22] (Error: 1.75)
Progression : [ 9;22] (Error: 1.44444)
Progression : [10;22] (Error: 1.2)
Answer: 3
move(c,table,0) move(i,table,0) move(k,table,0) move(b,table,1) move(h,table
,1) move(j,table,1) move(c,i,2) move(e,j,2) move(k,g,2) move(a,e,3) move(c,k
,3) move(d,table,3) move(b,c,4) move(n,a,4) move(d,n,5) move(i,d,6) move(h,i
,7) move(l,b,7) move(10,h,8) move(o,10,9)
Optimization: 20
Progression : [11;20] (Error: 0.818182)
Progression : [12;20] (Error: 0.666667)
Progression : [13;20] (Error: 0.538462)
Progression : [14;20] (Error: 0.428571)
```

```
Optimization: 20
Progression : [11;20] (Error: 0.818182)
Progression : [12;20] (Error: 0.666667)
Progression : [13;20] (Error: 0.538462)
Progression : [14;20] (Error: 0.428571)
Answer: 4
move(i,table,0) move(k,c,0) move(h,table,1) move(j,table,1) move(h,e,2) move
(k,g,2) move(c,k,3) move(h,table,3) move(b,c,4) move(e,j,4) move(i,h,4) move
(a,e,5) move(d,table,5) move(i,d,6) move(l,b,6) move(n,a,6) move(h,i,7) move
(10,h,8) move(o,10,9)
Optimization: 19
Progression : [15;19] (Error: 0.266667)
Answer: 5
move(i,table,0) move(k,c,0) move(h,table,1) move(j,table,1) move(k,i,1) move
(c,j,2) move(k,g,2) move(c,k,3) move(b,c,4) move(e,j,4) move(a,e,5) move(d,t
able,5) move(i,d,6) move(l,b,6) move(n,a,6) move(h,i,7) move(10,h,8) move(o,
10,9)
Optimization: 18
Answer: 6
move(i,table,0) move(h,table,1) move(k,g,2) move(c,k,3) move(j,table,3) move
(b,c,4) move(e,j,4) move(a,e,5) move(d,table,5) move(i,d,6) move(n,a,6) move
(h,i,7) move(l,b,7) move(10,h,8) move(o,10,9)
Optimization: 15
OPTIMUM FOUND

Models       : 6
  Optimum    : yes
Optimization : 15
Calls        : 1
Time         : 0.816s (Solving: 0.08s 1st Model: 0.01s Unsat: 0.00s)
CPU Time     : 0.891s
Threads      : 4        (Winner: 1)
```

Answer to Questions

What is the least number of actions when maxstep m is 8, 9, and 10?

m	least number of actions
8	18
9	16
10	15