Abstract: In this project we learned about state space problems and how to solve them. This mainly being done through the move predicates and the valid predicate. I learned that changing where a predicate is in prolog can change the order of execution, I learned this when attempting to get my code to work on 5 disk and seeing that it caused a infinite loop, I then moved the order of my predicates and was able t solve the 5 disk problem but the smaller disk problems take longer.

Task 1: Problem Contemplation - Towers of Hanoi

The three peg/ three tower problem:
I = { ((S M L) ( ) ( )) }
G = { (( ) ( ) (S M L)) }
O = {M12, M13, M21, M23, M31, M32}, where
– M12 - move a disk from peg 1 to peg 2
– M13 - move a disk from peg 1 to peg 3
– M21 - move a disk from peg 2 to peg 1
– M23 - move a disk from peg 2 to peg 3
– M31 - move a disk from peg 3 to peg 1
– M32 - move a disk from peg 3 to peg 2

The three peg/ three tower problem:
I = { ((S M L H) ( ) ( )) }
G = { (( ) ( ) (S M L H)) }
O = {M12, M13, M21, M23, M31, M32}, where
– M12 - move a disk from peg 1 to peg 2
– M13 - move a disk from peg 1 to peg 3
– M21 - move a disk from peg 2 to peg 1
– M23 - move a disk from peg 2 to peg 3
– M31 - move a disk from peg 3 to peg 1
– M32 - move a disk from peg 3 to peg 2

The three peg/ three tower problem:
I = { ((T S M L H) ( ) ( )) }
G = { (( ) ( ) (T S M L H)) }
O = {M12, M13, M21, M23, M31, M32}, where
– M12 - move a disk from peg 1 to peg 2
– M13 - move a disk from peg 1 to peg 3
– M21 - move a disk from peg 2 to peg 1
– M23 - move a disk from peg 2 to peg 3
– M31 - move a disk from peg 3 to peg 1
– M32 - move a disk from peg 3 to peg 2

Task 2: Code Contemplation

4 redacted statements to complete (I wasn’t sure what I would write for this section).

State operators: changing one state to another state

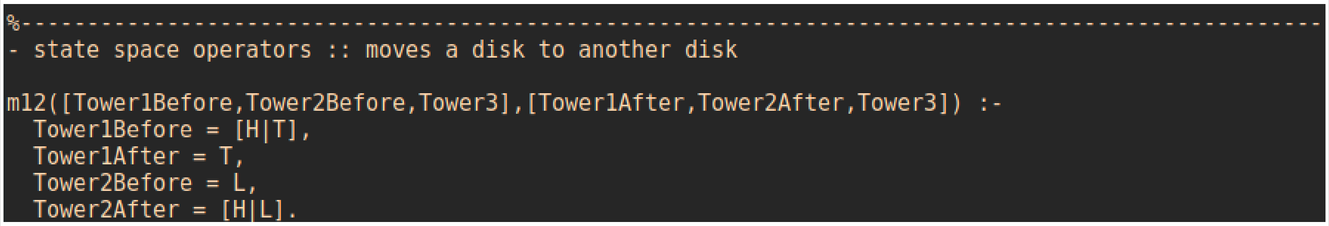
Valid_state: Checking if state is within constraints

Write_solution: expands moves into a sentence describing move

unit test

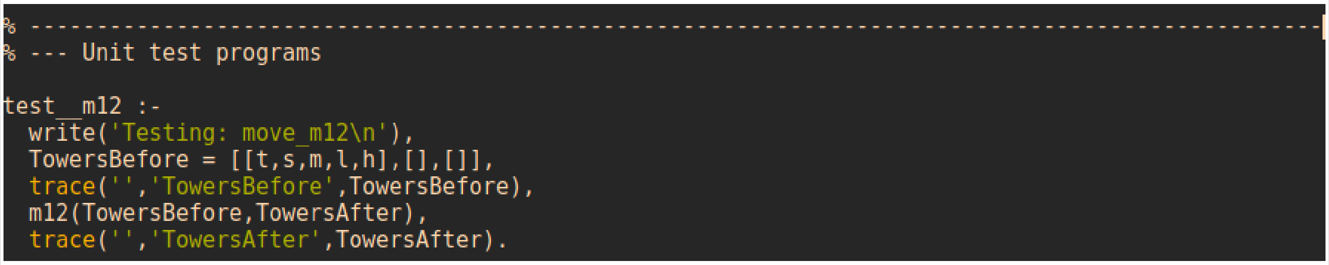
Task 3: One Move Predicate and a Unit Test

State Space Code:



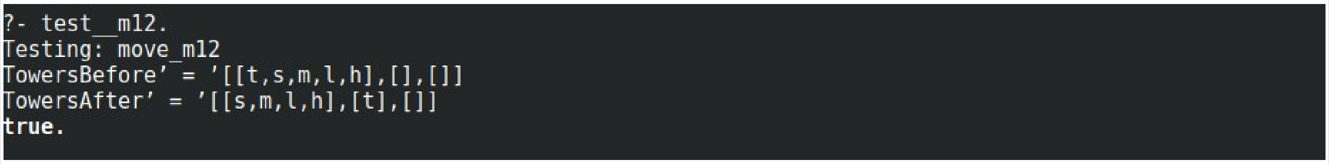
```
%-------------------------------------------------------------------------------------------
- state space operators :: moves a disk to another disk

m12([Tower1Before,Tower2Before,Tower3],[Tower1After,Tower2After,Tower3]) :-
  Tower1Before = [H|T],
  Tower1After = T,
  Tower2Before = L,
  Tower2After = [H|L].
```

Unit Test:



```
% -----------------------------------------------------------------------------------------
% --- Unit test programs

test__m12 :-
  write('Testing: move_m12\n'),
  TowersBefore = [[t,s,m,l,h],[],[]],
  trace('','TowersBefore',TowersBefore),
  m12(TowersBefore,TowersAfter),
  trace('','TowersAfter',TowersAfter).
```

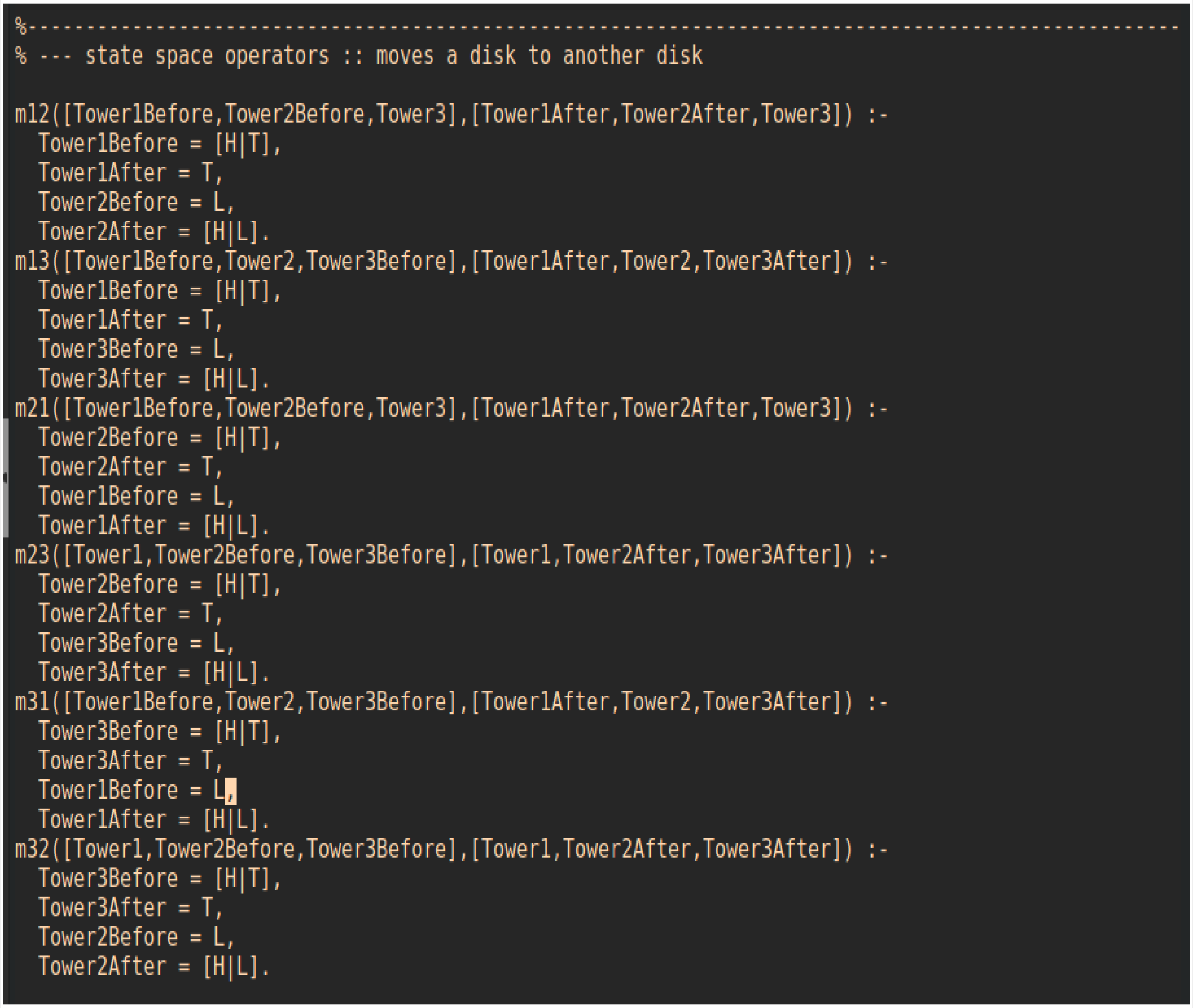
Demo:



```
?- test__m12.
Testing: move_m12
TowersBefore’ = ’[[t,s,m,l,h],[],[]]
TowersAfter’ = ’[[s,m,l,h],[t],[]]
true.
```

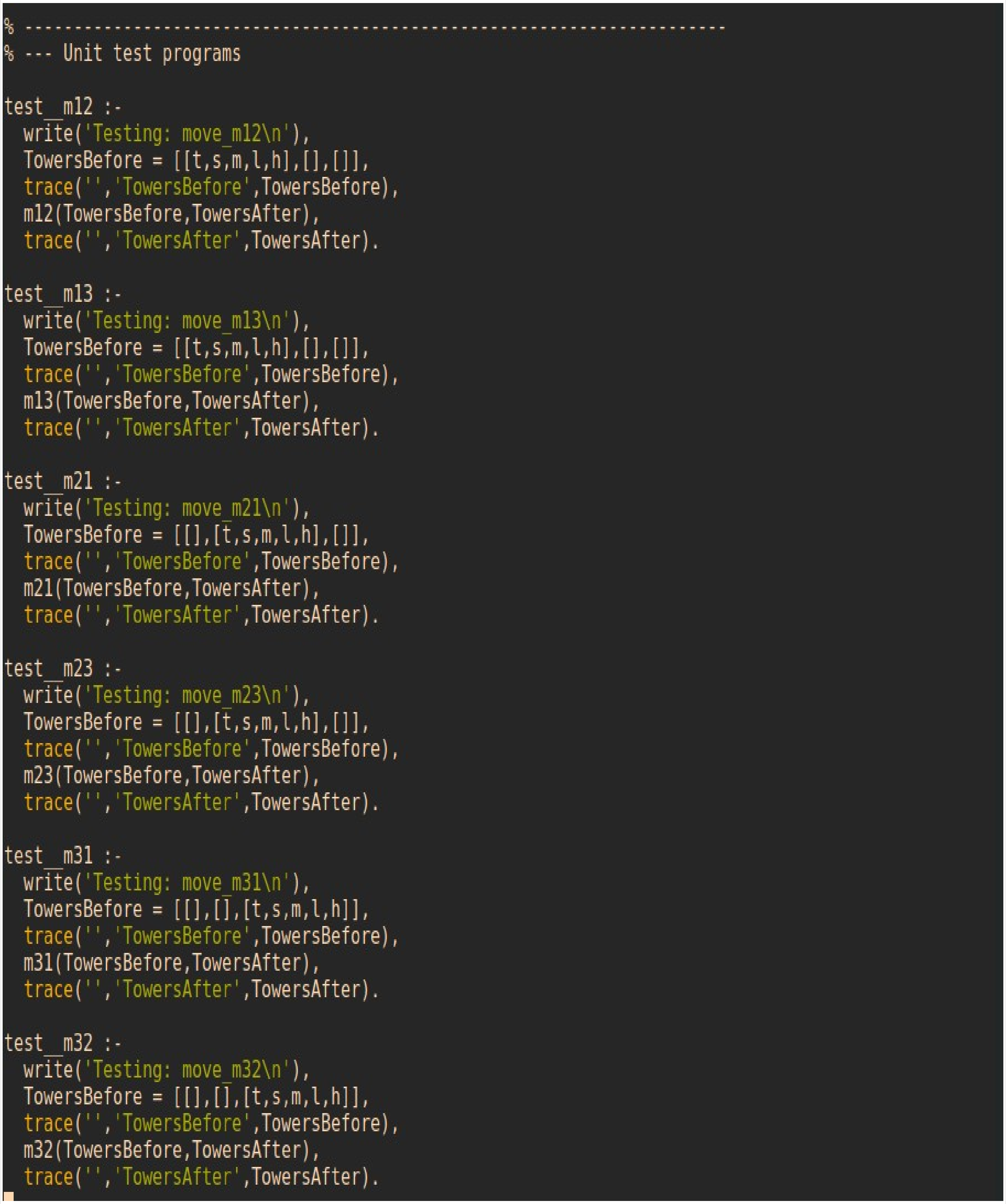
Task 4: The Remaining Five Move Predicates and a Unit Tests

State Space Operator Code:



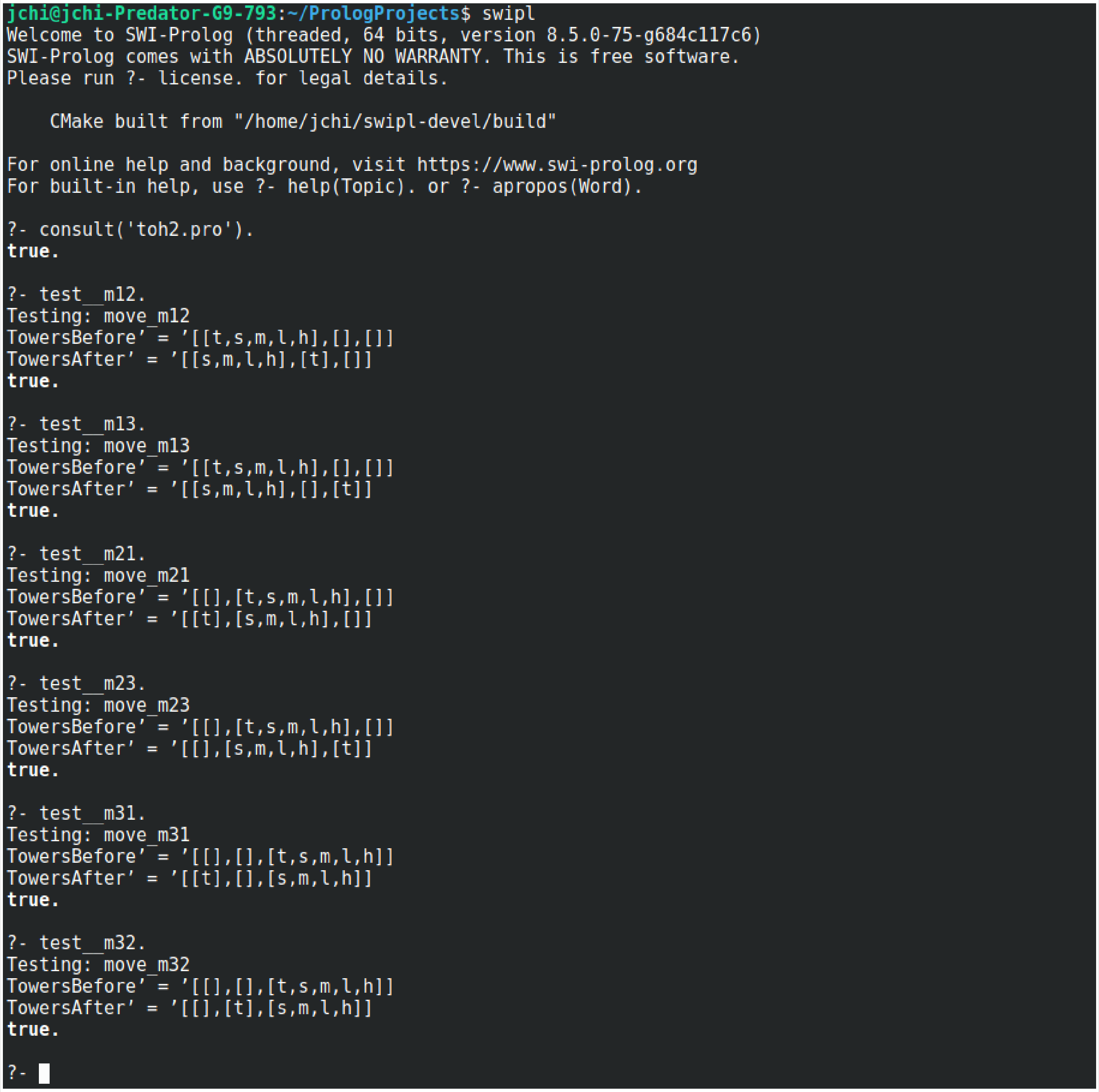
```
%-------------------------------------------------------------------------------------------------------
% --- state space operators :: moves a disk to another disk

m12([Tower1Before,Tower2Before,Tower3],[Tower1After,Tower2After,Tower3]) :-
  Tower1Before = [H|T],
  Tower1After = T,
  Tower2Before = L,
  Tower2After = [H|L].
m13([Tower1Before,Tower2,Tower3Before],[Tower1After,Tower2,Tower3After]) :-
  Tower1Before = [H|T],
  Tower1After = T,
  Tower3Before = L,
  Tower3After = [H|L].
m21([Tower1Before,Tower2Before,Tower3],[Tower1After,Tower2After,Tower3]) :-
  Tower2Before = [H|T],
  Tower2After = T,
  Tower1Before = L,
  Tower1After = [H|L].
m23([Tower1,Tower2Before,Tower3Before],[Tower1,Tower2After,Tower3After]) :-
  Tower2Before = [H|T],
  Tower2After = T,
  Tower3Before = L,
  Tower3After = [H|L].
m31([Tower1Before,Tower2,Tower3Before],[Tower1After,Tower2,Tower3After]) :-
  Tower3Before = [H|T],
  Tower3After = T,
  Tower1Before = L,
  Tower1After = [H|L].
m32([Tower1,Tower2Before,Tower3Before],[Tower1,Tower2After,Tower3After]) :-
  Tower3Before = [H|T],
  Tower3After = T,
  Tower2Before = L,
  Tower2After = [H|L].
```

Unit Test Code:



```
% -----------------------------------------------------------------------
% --- Unit test programs

test__m12 :-
  write('Testing: move_m12\n'),
  TowersBefore = [[t,s,m,l,h],[],[]],
  trace('','TowersBefore',TowersBefore),
  m12(TowersBefore,TowersAfter),
  trace('','TowersAfter',TowersAfter).

test__m13 :-
  write('Testing: move_m13\n'),
  TowersBefore = [[t,s,m,l,h],[],[]],
  trace('','TowersBefore',TowersBefore),
  m13(TowersBefore,TowersAfter),
  trace('','TowersAfter',TowersAfter).

test__m21 :-
  write('Testing: move_m21\n'),
  TowersBefore = [[],[t,s,m,l,h],[]],
  trace('','TowersBefore',TowersBefore),
  m21(TowersBefore,TowersAfter),
  trace('','TowersAfter',TowersAfter).

test__m23 :-
  write('Testing: move_m23\n'),
  TowersBefore = [[],[t,s,m,l,h],[]],
  trace('','TowersBefore',TowersBefore),
  m23(TowersBefore,TowersAfter),
  trace('','TowersAfter',TowersAfter).

test__m31 :-
  write('Testing: move_m31\n'),
  TowersBefore = [[],[],[t,s,m,l,h]],
  trace('','TowersBefore',TowersBefore),
  m31(TowersBefore,TowersAfter),
  trace('','TowersAfter',TowersAfter).

test__m32 :-
  write('Testing: move_m32\n'),
  TowersBefore = [[],[],[t,s,m,l,h]],
  trace('','TowersBefore',TowersBefore),
  m32(TowersBefore,TowersAfter),
  trace('','TowersAfter',TowersAfter).
```

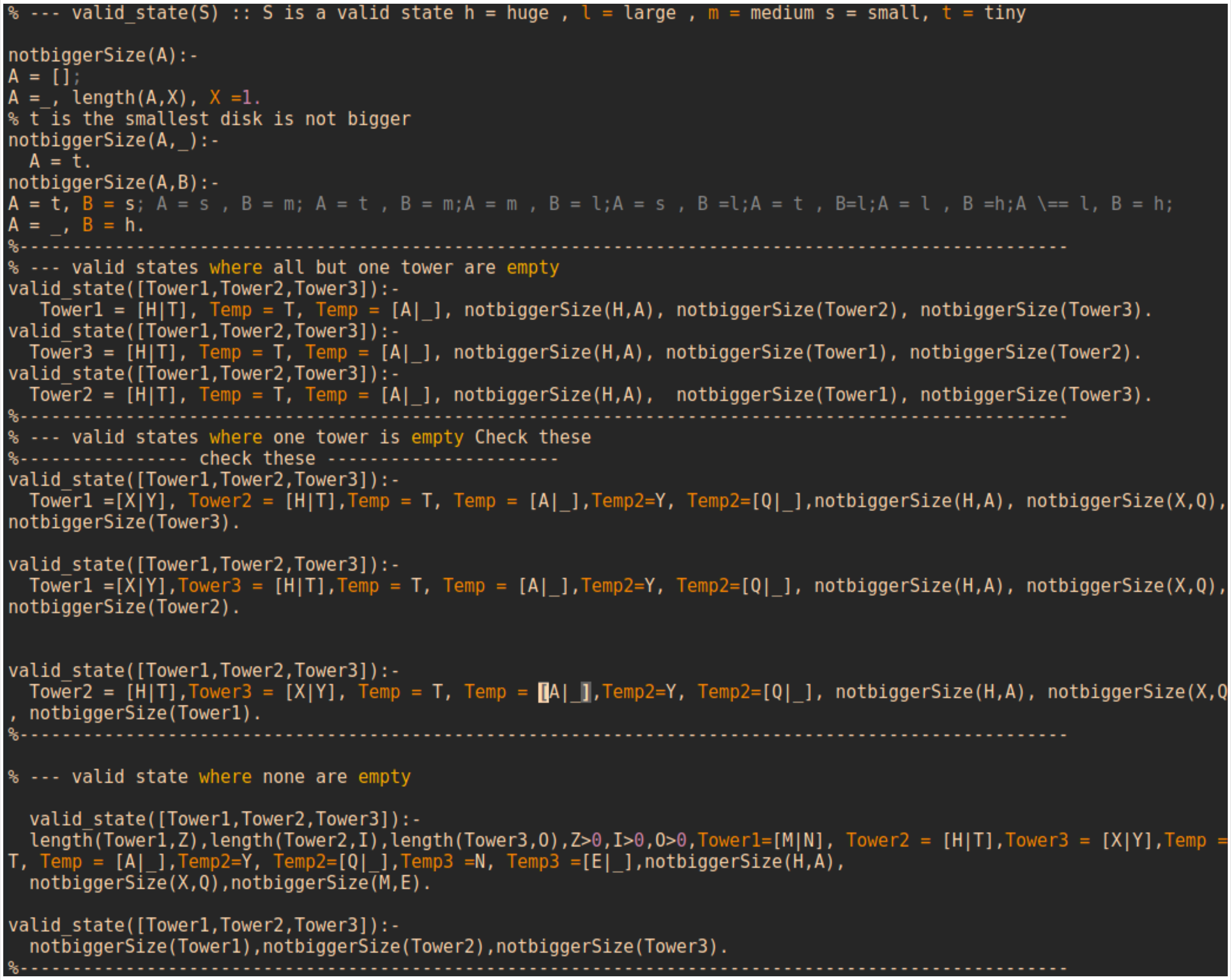
Demo:



```
jchi@jchi-Predator-G9-793:~/PrologProjects$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.5.0-75-g684c117c6)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

    CMake built from "/home/jchi/swipl-devel/build"

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('toh2.pro').
true.

?- test__m12.
Testing: move_m12
TowersBefore’ = ’[[t,s,m,l,h],[],[]]
TowersAfter’ = ’[[s,m,l,h],[t],[]]
true.

?- test__m13.
Testing: move_m13
TowersBefore’ = ’[[t,s,m,l,h],[],[]]
TowersAfter’ = ’[[s,m,l,h],[],[t]]
true.

?- test__m21.
Testing: move_m21
TowersBefore’ = ’[[],[t,s,m,l,h],[]]
TowersAfter’ = ’[[t],[s,m,l,h],[]]
true.

?- test__m23.
Testing: move_m23
TowersBefore’ = ’[[],[t,s,m,l,h],[]]
TowersAfter’ = ’[[],[s,m,l,h],[t]]
true.

?- test__m31.
Testing: move_m31
TowersBefore’ = ’[[],[],[t,s,m,l,h]]
TowersAfter’ = ’[[t],[],[s,m,l,h]]
true.

?- test__m32.
Testing: move_m32
TowersBefore’ = ’[[],[],[t,s,m,l,h]]
TowersAfter’ = ’[[],[t],[s,m,l,h]]
true.

?- 
```

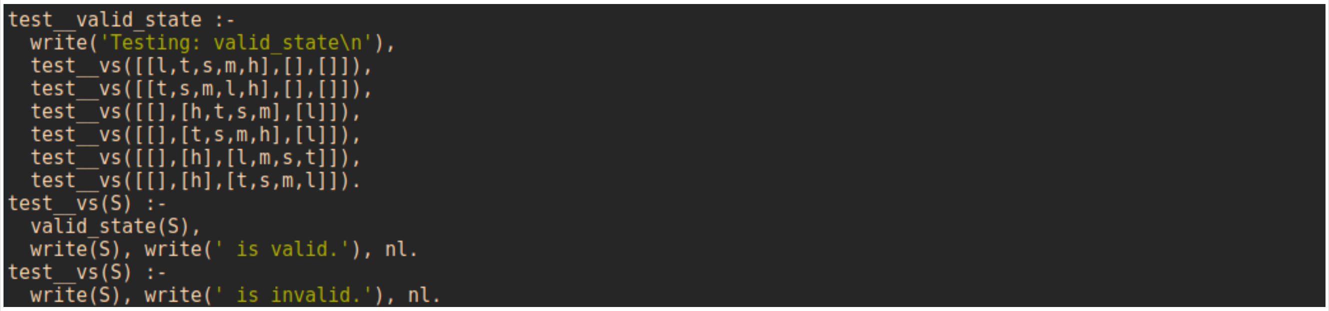
Task 5: Valid State Predicate and Unit Test

Valid State Code:



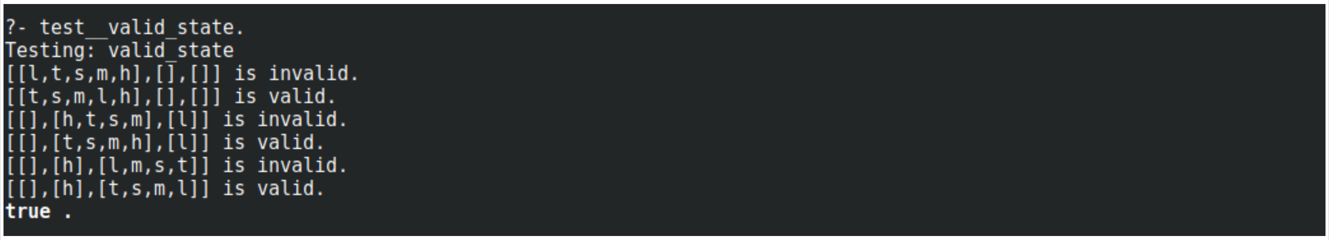
```
% --- valid_state(S) :: S is a valid state h = huge , l = large , m = medium s = small, t = tiny

notbiggerSize(A):-
A = [];
A =_, length(A,X), X =1.
% t is the smallest disk is not bigger
notbiggerSize(A,_):-
  A = t.
notbiggerSize(A,B):-
A = t, B = s; A = s , B = m; A = t , B = m;A = m , B = l;A = s , B =l;A = t , B=l;A = l , B =h;A \== l, B = h;
A = _, B = h.
%-------------------------------------------------------------------------------------------------
% --- valid states where all but one tower are empty
valid_state([Tower1,Tower2,Tower3]):-
   Tower1 = [H|T], Temp = T, Temp = [A|_], notbiggerSize(H,A), notbiggerSize(Tower2), notbiggerSize(Tower3).
valid_state([Tower1,Tower2,Tower3]):-
  Tower3 = [H|T], Temp = T, Temp = [A|_], notbiggerSize(H,A), notbiggerSize(Tower1), notbiggerSize(Tower2).
valid_state([Tower1,Tower2,Tower3]):-
  Tower2 = [H|T], Temp = T, Temp = [A|_], notbiggerSize(H,A),  notbiggerSize(Tower1), notbiggerSize(Tower3).
%-------------------------------------------------------------------------------------------------
% --- valid states where one tower is empty Check these
%--------------- check these ---------------------
valid_state([Tower1,Tower2,Tower3]):-
  Tower1 =[X|Y], Tower2 = [H|T],Temp = T, Temp = [A|_],Temp2=Y, Temp2=[Q|_],notbiggerSize(H,A), notbiggerSize(X,Q),
notbiggerSize(Tower3).

valid_state([Tower1,Tower2,Tower3]):-
  Tower1 =[X|Y],Tower3 = [H|T],Temp = T, Temp = [A|_],Temp2=Y, Temp2=[Q|_], notbiggerSize(H,A), notbiggerSize(X,Q),
notbiggerSize(Tower2).


valid_state([Tower1,Tower2,Tower3]):-
  Tower2 = [H|T],Tower3 = [X|Y], Temp = T, Temp = [A|_],Temp2=Y, Temp2=[Q|_], notbiggerSize(H,A), notbiggerSize(X,Q
, notbiggerSize(Tower1).
%-------------------------------------------------------------------------------------------------

% --- valid state where none are empty

  valid_state([Tower1,Tower2,Tower3]):-
  length(Tower1,Z),length(Tower2,I),length(Tower3,O),Z>0,I>0,O>0,Tower1=[M|N], Tower2 = [H|T],Tower3 = [X|Y],Temp =
T, Temp = [A|_],Temp2=Y, Temp2=[Q|_],Temp3 =N, Temp3 =[E|_],notbiggerSize(H,A),
  notbiggerSize(X,Q),notbiggerSize(M,E).

valid_state([Tower1,Tower2,Tower3]):-
  notbiggerSize(Tower1),notbiggerSize(Tower2),notbiggerSize(Tower3).
%-------------------------------------------------------------------------------------------------
```

Unit Test Code:



```
test__valid_state :-
  write('Testing: valid_state\n'),
  test__vs([[l,t,s,m,h],[],[]]),
  test__vs([[t,s,m,l,h],[],[]]),
  test__vs([[],[h,t,s,m],[l]]),
  test__vs([[],[t,s,m,h],[l]]),
  test__vs([[],[h],[l,m,s,t]]),
  test__vs([[],[h],[t,s,m,l]]).
test__vs(S) :-
  valid_state(S),
  write(S), write(' is valid.'), nl.
test__vs(S) :-
  write(S), write(' is invalid.'), nl.
```

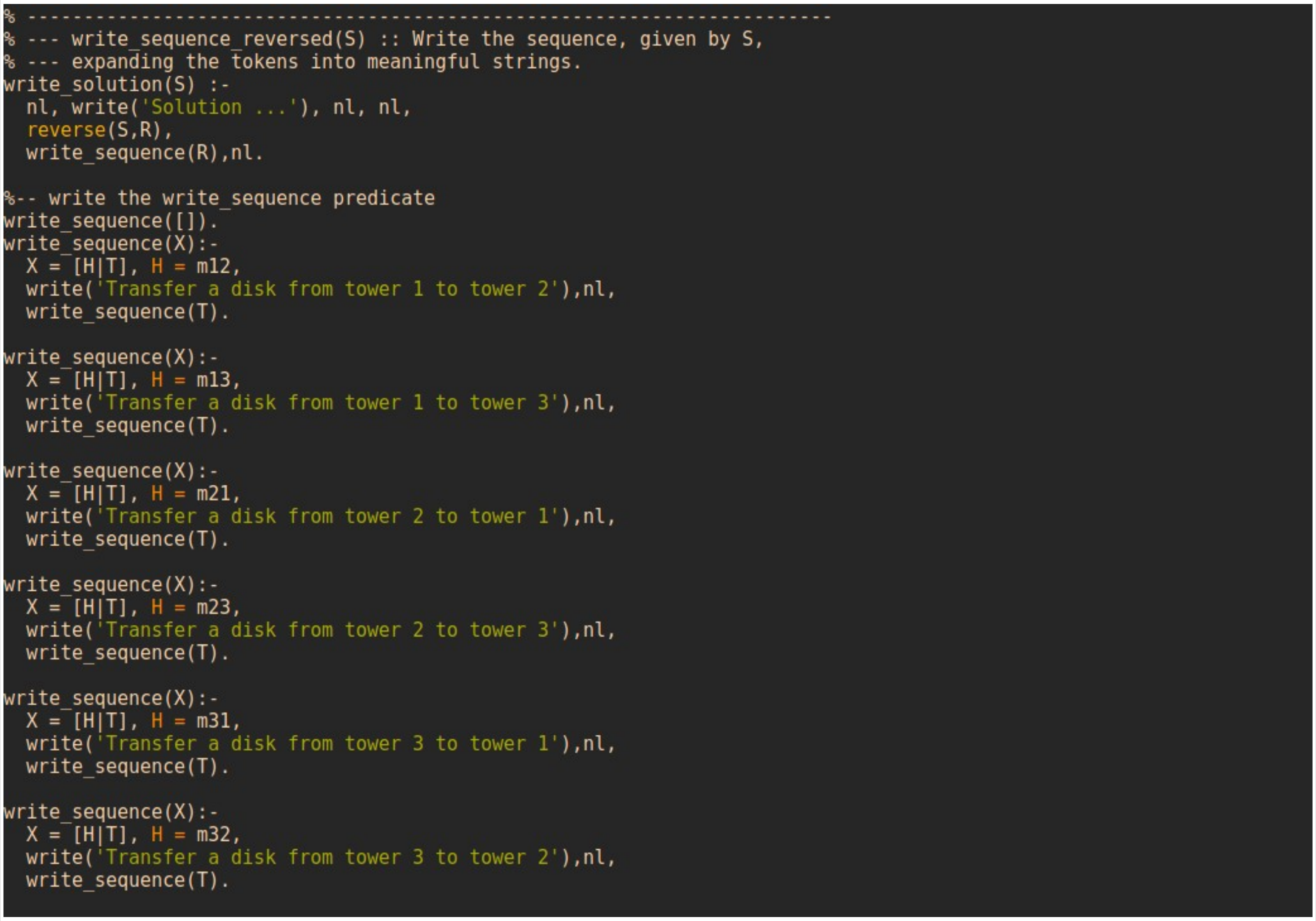
Demo:



```
?- test__valid_state.
Testing: valid_state
[[l,t,s,m,h],[],[]] is invalid.
[[t,s,m,l,h],[],[]] is valid.
[[],[h,t,s,m],[l]] is invalid.
[[],[t,s,m,h],[l]] is valid.
[[],[h],[l,m,s,t]] is invalid.
[[],[h],[t,s,m,l]] is valid.
true .
```

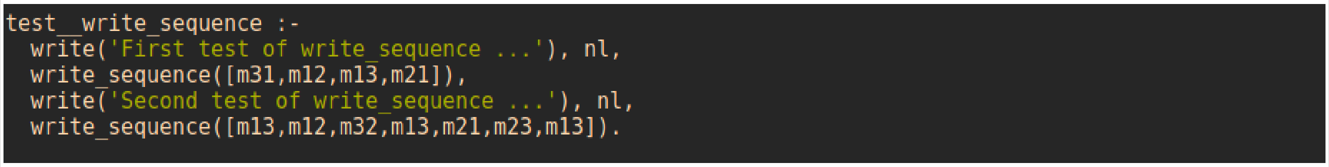
Task 6: Defining the write sequence predicate

Write Sequence Code:



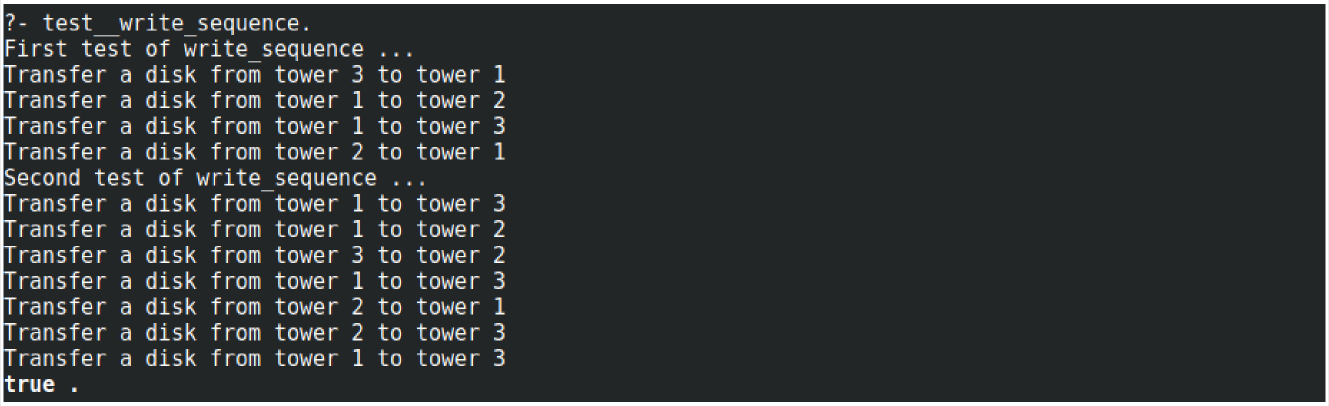
```
% -----------------------------------------------------------------------
% --- write_sequence_reversed(S) :: Write the sequence, given by S,
% --- expanding the tokens into meaningful strings.
write_solution(S) :-
  nl, write('Solution ...'), nl, nl,
  reverse(S,R),
  write_sequence(R),nl.

%-- write the write_sequence predicate
write_sequence([]).
write_sequence(X):-
  X = [H|T], H = m12,
  write('Transfer a disk from tower 1 to tower 2'),nl,
  write_sequence(T).

write_sequence(X):-
  X = [H|T], H = m13,
  write('Transfer a disk from tower 1 to tower 3'),nl,
  write_sequence(T).

write_sequence(X):-
  X = [H|T], H = m21,
  write('Transfer a disk from tower 2 to tower 1'),nl,
  write_sequence(T).

write_sequence(X):-
  X = [H|T], H = m23,
  write('Transfer a disk from tower 2 to tower 3'),nl,
  write_sequence(T).

write_sequence(X):-
  X = [H|T], H = m31,
  write('Transfer a disk from tower 3 to tower 1'),nl,
  write_sequence(T).

write_sequence(X):-
  X = [H|T], H = m32,
  write('Transfer a disk from tower 3 to tower 2'),nl,
  write_sequence(T).
```

Unit Test Code:



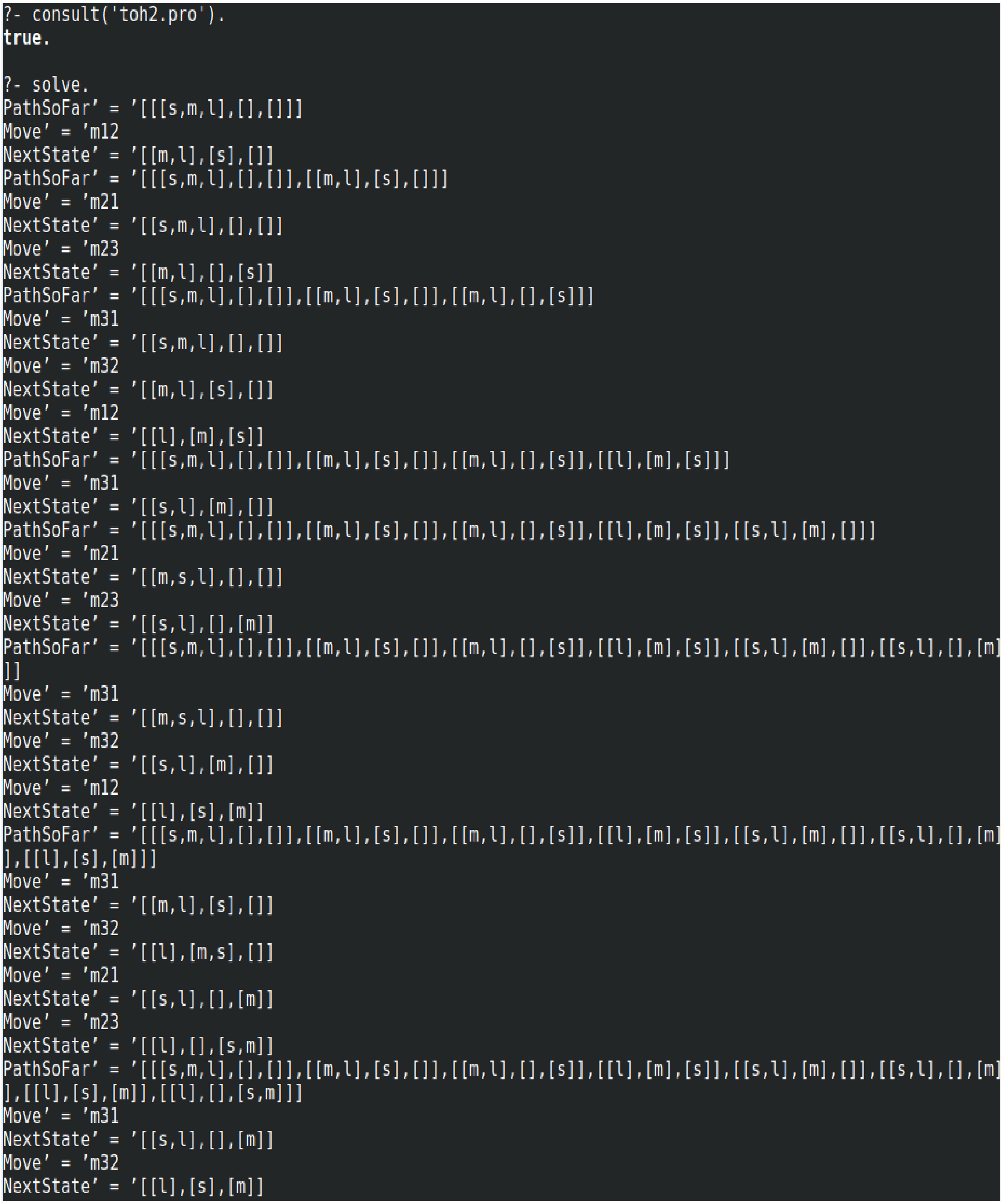
```
test__write_sequence :-
  write('First test of write_sequence ...'), nl,
  write_sequence([m31,m12,m13,m21]),
  write('Second test of write_sequence ...'), nl,
  write_sequence([m13,m12,m32,m13,m21,m23,m13]).
```

Demo:

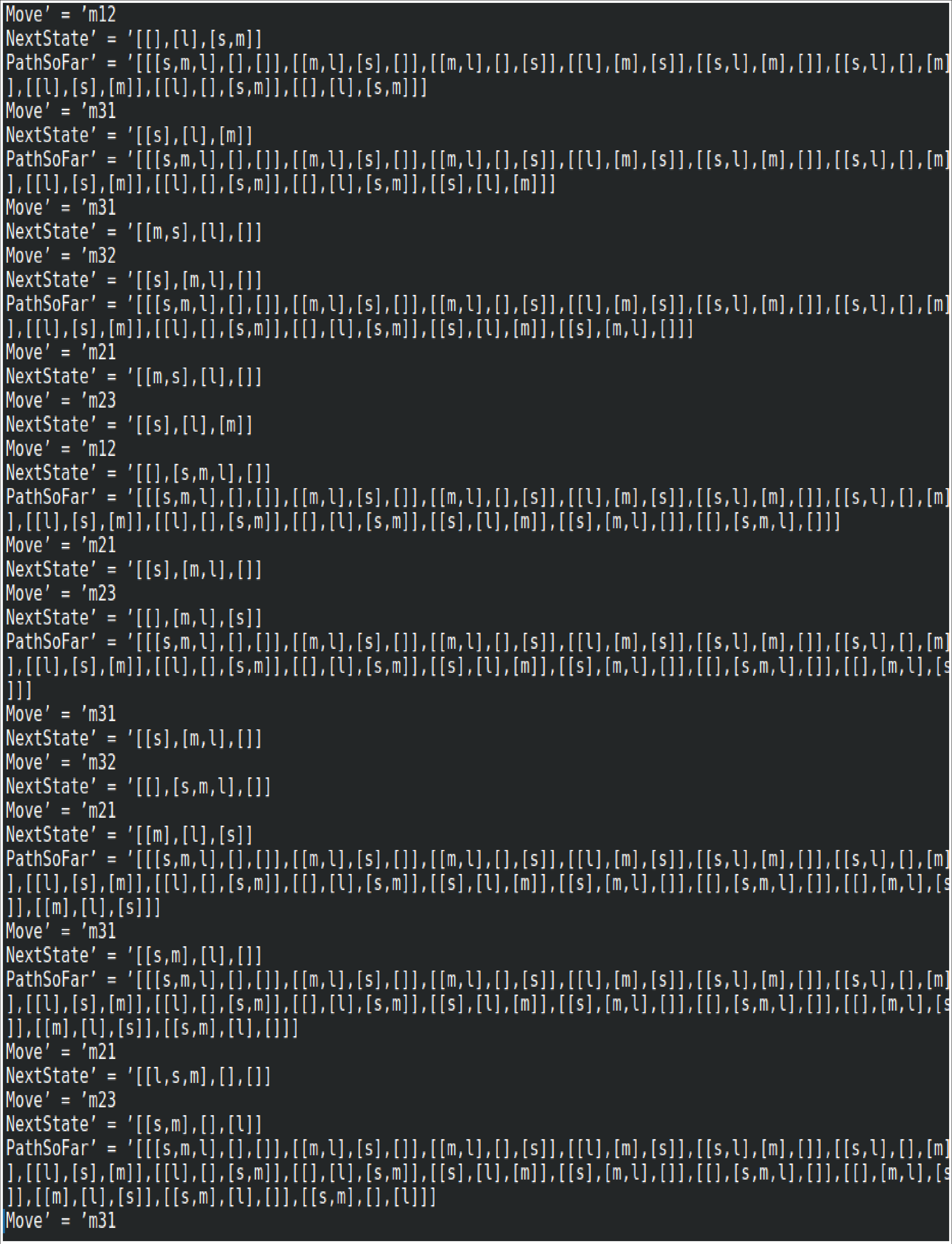


```
?- test__write_sequence.
First test of write_sequence ...
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 1 to tower 3
Transfer a disk from tower 2 to tower 1
Second test of write_sequence ...
Transfer a disk from tower 1 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 2
Transfer a disk from tower 1 to tower 3
Transfer a disk from tower 2 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 3
true .
```

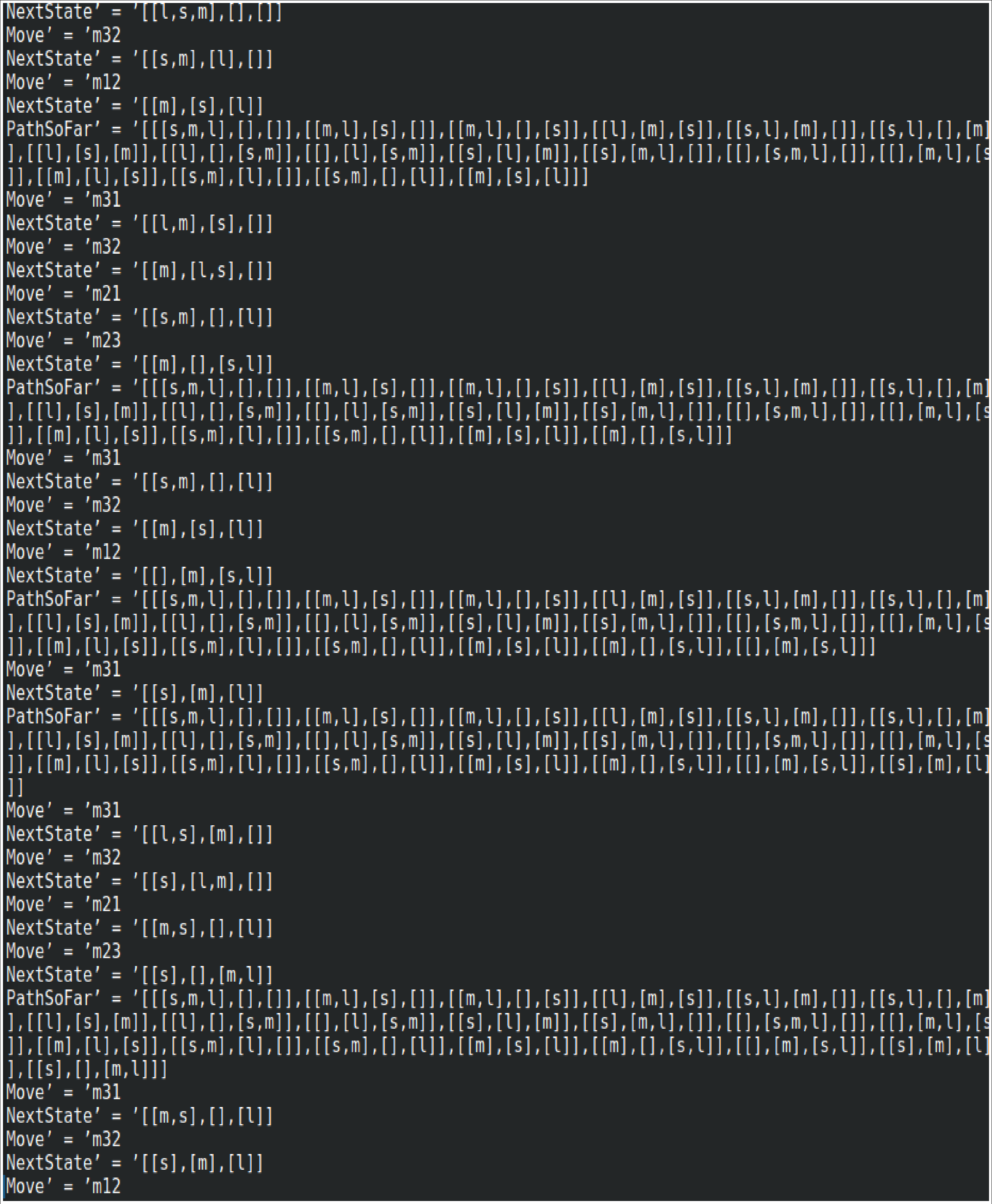
Task 7: Run the program to solve the 3 disk problem



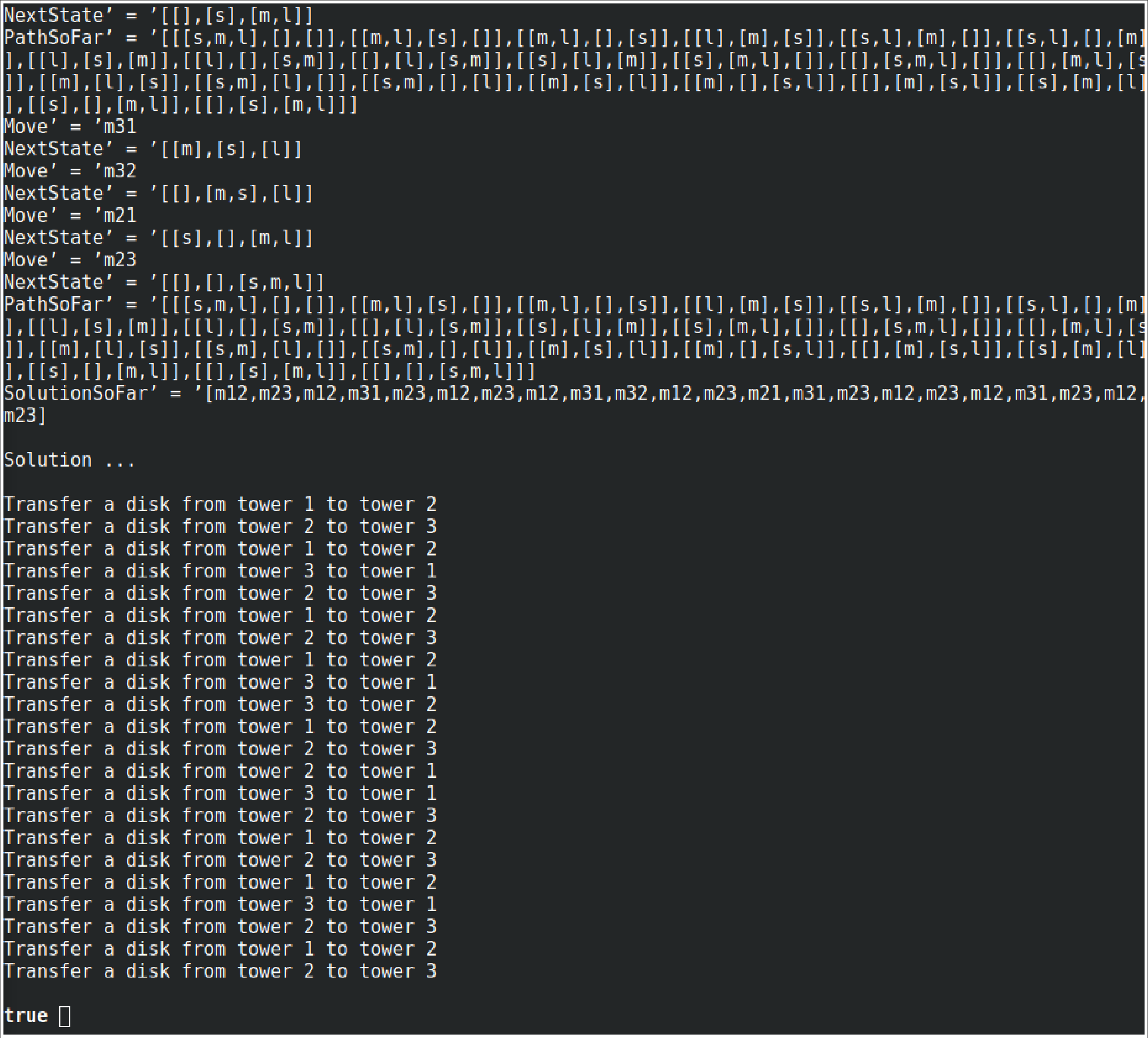
```
?- consult('toh2.pro').
true.

?- solve.
PathSoFar’ = ’[[[s,m,l],[],[]]]
Move’ = ’m12
NextState’ = ’[[m,l],[s],[]]
PathSoFar’ = ’[[[s,m,l],[],[]],[[m,l],[s],[]]]
Move’ = ’m21
NextState’ = ’[[s,m,l],[],[]]
Move’ = ’m23
NextState’ = ’[[m,l],[],[s]]
PathSoFar’ = ’[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]]]
Move’ = ’m31
NextState’ = ’[[s,m,l],[],[]]
Move’ = ’m32
NextState’ = ’[[m,l],[s],[]]
Move’ = ’m12
NextState’ = ’[[l],[m],[s]]
PathSoFar’ = ’[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]]]
Move’ = ’m31
NextState’ = ’[[s,l],[m],[]]
PathSoFar’ = ’[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]]]
Move’ = ’m21
NextState’ = ’[[m,s,l],[],[]]
Move’ = ’m23
NextState’ = ’[[s,l],[],[m]]
PathSoFar’ = ’[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
]]
Move’ = ’m31
NextState’ = ’[[m,s,l],[],[]]
Move’ = ’m32
NextState’ = ’[[s,l],[m],[]]
Move’ = ’m12
NextState’ = ’[[l],[s],[m]]
PathSoFar’ = ’[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]]]
Move’ = ’m31
NextState’ = ’[[m,l],[s],[]]
Move’ = ’m32
NextState’ = ’[[l],[m,s],[]]
Move’ = ’m21
NextState’ = ’[[s,l],[],[m]]
Move’ = ’m23
NextState’ = ’[[l],[],[s,m]]
PathSoFar’ = ’[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]]]
Move’ = ’m31
NextState’ = ’[[s,l],[],[m]]
Move’ = ’m32
NextState’ = ’[[l],[s],[m]]
```



```
Move' = 'm12
NextState' = '[[],[l],[s,m]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]]]
Move' = 'm31
NextState' = '[[s],[l],[m]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]]]
Move' = 'm31
NextState' = '[[m,s],[l],[]]
Move' = 'm32
NextState' = '[[s],[m,l],[]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]]]
Move' = 'm21
NextState' = '[[m,s],[l],[]]
Move' = 'm23
NextState' = '[[s],[l],[m]]
Move' = 'm12
NextState' = '[[],[s,m,l],[]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]]]
Move' = 'm21
NextState' = '[[s],[m,l],[]]
Move' = 'm23
NextState' = '[[],[m,l],[s]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]]]
Move' = 'm31
NextState' = '[[s],[m,l],[]]
Move' = 'm32
NextState' = '[[],[s,m,l],[]]
Move' = 'm21
NextState' = '[[m],[l],[s]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]]]
Move' = 'm31
NextState' = '[[s,m],[l],[]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]],[[s,m],[l],[]]]
Move' = 'm21
NextState' = '[[l,s,m],[],[]]
Move' = 'm23
NextState' = '[[s,m],[],[l]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]],[[s,m],[l],[]],[[s,m],[],[l]]]
Move' = 'm31
```



```
NextState' = '[[l,s,m],[],[]]
Move' = 'm32
NextState' = '[[s,m],[l],[]]
Move' = 'm12
NextState' = '[[m],[s],[l]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]]]
Move' = 'm31
NextState' = '[[l,m],[s],[]]
Move' = 'm32
NextState' = '[[m],[l,s],[]]
Move' = 'm21
NextState' = '[[s,m],[],[l]]
Move' = 'm23
NextState' = '[[m],[],[s,l]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]],[[m],[],[s,l]]]
Move' = 'm31
NextState' = '[[s,m],[],[l]]
Move' = 'm32
NextState' = '[[m],[s],[l]]
Move' = 'm12
NextState' = '[[],[m],[s,l]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]],[[m],[],[s,l]],[[],[m],[s,l]]]
Move' = 'm31
NextState' = '[[s],[m],[l]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]],[[m],[],[s,l]],[[],[m],[s,l]],[[s],[m],[l]
]]
Move' = 'm31
NextState' = '[[l,s],[m],[]]
Move' = 'm32
NextState' = '[[s],[l,m],[]]
Move' = 'm21
NextState' = '[[m,s],[],[l]]
Move' = 'm23
NextState' = '[[s],[],[m,l]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]],[[m],[],[s,l]],[[],[m],[s,l]],[[s],[m],[l]
],[[s],[],[m,l]]]
Move' = 'm31
NextState' = '[[m,s],[],[l]]
Move' = 'm32
NextState' = '[[s],[m],[l]]
Move' = 'm12
```



```
NextState' = '[[],[s],[m,l]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]],[[m],[],[s,l]],[[],[m],[s,l]],[[s],[m],[l]
],[[s],[],[m,l]],[[],[s],[m,l]]]
Move' = 'm31
NextState' = '[[m],[s],[l]]
Move' = 'm32
NextState' = '[[],[m,s],[l]]
Move' = 'm21
NextState' = '[[s],[],[m,l]]
Move' = 'm23
NextState' = '[[],[],[s,m,l]]
PathSoFar' = '[[[s,m,l],[],[]],[[m,l],[s],[]],[[m,l],[],[s]],[[l],[m],[s]],[[s,l],[m],[]],[[s,l],[],[m]
],[[l],[s],[m]],[[l],[],[s,m]],[[],[l],[s,m]],[[s],[l],[m]],[[s],[m,l],[]],[[],[s,m,l],[]],[[],[m,l],[s
]],[[m],[l],[s]],[[s,m],[l],[]],[[s,m],[],[l]],[[m],[s],[l]],[[m],[],[s,l]],[[],[m],[s,l]],[[s],[m],[l]
],[[s],[],[m,l]],[[],[s],[m,l]],[[],[],[s,m,l]]]
SolutionSoFar' = '[m12,m23,m12,m31,m23,m12,m23,m12,m31,m32,m12,m23,m21,m31,m23,m12,m23,m12,m31,m23,m12,
m23]

Solution ...

Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 3 to tower 2
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 2 to tower 1
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3

true 
```

1. What was the length of your program’s solution to the three disk problem?

21

2. What is the length of the shortest solution to the three disk problem?

7 moves

3. How do you account for the discrepancy?

The order of my make move predicate causes it to move extra moves.

Task 8: Run the program to solve the 4 disk problem

```
jchi@jchi-Predator-G9-793:~/PrologProjects$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.5.0-75-
g684c117c6)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.

Please run ?- license. for legal details.

   CMake built from "/home/jchi/swipl-devel/build"

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('toh2.pro').
true.

?- solve.
PathSoFar’ = ’[[[s,m,l,h],[],[]]]
Move’ = ’m12
NextState’ = ’[[m,l,h],[s],[]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]]]
Move’ = ’m21
NextState’ = ’[[s,m,l,h],[],[]]
Move’ = ’m23
NextState’ = ’[[m,l,h],[],[s]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]]]
Move’ = ’m31
NextState’ = ’[[s,m,l,h],[],[]]
Move’ = ’m32
NextState’ = ’[[m,l,h],[s],[]]
Move’ = ’m12
NextState’ = ’[[l,h],[m],[s]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]]]
Move’ = ’m31
NextState’ = ’[[s,l,h],[m],[]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]]]
Move’ = ’m21
NextState’ = ’[[m,s,l,h],[],[]]
```

```
Move’ = ’m23
NextState’ = ’[[s,l,h],[],[m]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]]]
Move’ = ’m31
NextState’ = ’[[m,s,l,h],[],[]]
Move’ = ’m32
NextState’ = ’[[s,l,h],[m],[]]
Move’ = ’m12
NextState’ = ’[[l,h],[s],[m]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]]]
Move’ = ’m31
NextState’ = ’[[m,l,h],[s],[]]
Move’ = ’m32
NextState’ = ’[[l,h],[m,s],[]]
Move’ = ’m21
NextState’ = ’[[s,l,h],[],[m]]
Move’ = ’m23
NextState’ = ’[[l,h],[],[s,m]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]]]
Move’ = ’m31
NextState’ = ’[[s,l,h],[],[m]]
Move’ = ’m32
NextState’ = ’[[l,h],[s],[m]]
Move’ = ’m12
NextState’ = ’[[h],[l],[s,m]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]]]
Move’ = ’m31
NextState’ = ’[[s,h],[l],[m]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]]]
Move’ = ’m31
NextState’ = ’[[m,s,h],[l],[]]
```

```
Move’ = ’m32
NextState’ = ’[[s,h],[m,l],[]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]]]
Move’ = ’m21
NextState’ = ’[[m,s,h],[l],[]]
Move’ = ’m23
NextState’ = ’[[s,h],[l],[m]]
Move’ = ’m12
NextState’ = ’[[h],[s,m,l],[]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]]]
Move’ = ’m21
NextState’ = ’[[s,h],[m,l],[]]
Move’ = ’m23
NextState’ = ’[[h],[m,l],[s]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]]]
Move’ = ’m31
NextState’ = ’[[s,h],[m,l],[]]
Move’ = ’m32
NextState’ = ’[[h],[s,m,l],[]]
Move’ = ’m21
NextState’ = ’[[m,h],[l],[s]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]]]
Move’ = ’m31
NextState’ = ’[[s,m,h],[l],[]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]]]
Move’ = ’m21
NextState’ = ’[[l,s,m,h],[],[]]
Move’ = ’m23
NextState’ = ’[[s,m,h],[],[l]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]]]
Move’ = ’m31
NextState’ = ’[[l,s,m,h],[],[]]
Move’ = ’m32
NextState’ = ’[[s,m,h],[l],[]]
Move’ = ’m12
NextState’ = ’[[m,h],[s],[l]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]]]
Move’ = ’m31
NextState’ = ’[[l,m,h],[s],[]]
Move’ = ’m32
NextState’ = ’[[m,h],[l,s],[]]
Move’ = ’m21
NextState’ = ’[[s,m,h],[],[l]]
Move’ = ’m23
NextState’ = ’[[m,h],[],[s,l]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]]]
Move’ = ’m31
NextState’ = ’[[s,m,h],[],[l]]
Move’ = ’m32
```

```
NextState' = '[[m,h],[s],[l]]
Move' = 'm12
NextState' = '[[h],[m],[s,l]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]]]
Move' = 'm31
NextState' = '[[s,h],[m],[l]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]]]
Move' = 'm31
NextState' = '[[l,s,h],[m],[]]
Move' = 'm32
NextState' = '[[s,h],[l,m],[]]
Move' = 'm21
NextState' = '[[m,s,h],[],[l]]
Move' = 'm23
NextState' = '[[s,h],[],[m,l]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]]]
Move' = 'm31
NextState' = '[[m,s,h],[],[l]]
Move' = 'm32
NextState' = '[[s,h],[m],[l]]
Move' = 'm12
NextState' = '[[h],[s],[m,l]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]]]
Move' = 'm31
NextState' = '[[m,h],[s],[l]]
Move' = 'm32
NextState' = '[[h],[m,s],[l]]
Move' = 'm21
NextState' = '[[s,h],[],[m,l]]
Move' = 'm23
NextState' = '[[h],[],[s,m,l]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]]]
Move' = 'm31
NextState' = '[[s,h],[],[m,l]]
Move' = 'm32
NextState' = '[[h],[s],[m,l]]
Move' = 'm12
NextState' = '[[],[h],[s,m,l]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
]]
Move' = 'm31
NextState' = '[[s],[h],[m,l]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]]]
Move' = 'm31
NextState' = '[[m,s],[h],[l]]
Move' = 'm32
NextState' = '[[s],[m,h],[l]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]]]
Move' = 'm31
NextState' = '[[l,s],[m,h],[]]
Move' = 'm32
NextState' = '[[s],[l,m,h],[]]
Move' = 'm21
NextState' = '[[m,s],[h],[l]]
Move' = 'm23
NextState' = '[[s],[h],[m,l]]
Move' = 'm12
NextState' = '[[],[s,m,h],[l]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]]]
Move' = 'm31
NextState' = '[[l],[s,m,h],[]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
```

```
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]]]

Move' = 'm21
NextState' = '[[s,l],[m,h],[]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]]]
Move' = 'm21
NextState' = '[[m,s,l],[h],[]]
Move' = 'm23
NextState' = '[[s,l],[h],[m]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]]]
Move' = 'm31
NextState' = '[[m,s,l],[h],[]]
Move' = 'm32
NextState' = '[[s,l],[m,h],[]]
Move' = 'm21
NextState' = '[[h,s,l],[],[m]]
Move' = 'm23
NextState' = '[[s,l],[],[h,m]]
```

```
Move’ = ’m12
NextState’ = ’[[l],[s,h],[m]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]]]
Move’ = ’m31
NextState’ = ’[[m,l],[s,h],[]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]]]
Move’ = ’m21
NextState’ = ’[[s,m,l],[h],[]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]]]
Move’ = ’m21
NextState’ = ’[[h,s,m,l],[],[]]
Move’ = ’m23
```

```
NextState’ = ’[[s,m,l],[],[h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]]]
Move’ = ’m31
NextState’ = ’[[h,s,m,l],[],[]]
Move’ = ’m32
NextState’ = ’[[s,m,l],[h],[]]
Move’ = ’m12
NextState’ = ’[[m,l],[s],[h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]]]
Move’ = ’m31
NextState’ = ’[[h,m,l],[s],[]]
Move’ = ’m32
NextState’ = ’[[m,l],[h,s],[]]
Move’ = ’m21
NextState’ = ’[[s,m,l],[],[h]]
Move’ = ’m23
NextState’ = ’[[m,l],[],[s,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]]]
Move’ = ’m31
NextState’ = ’[[s,m,l],[],[h]]
Move’ = ’m32
NextState’ = ’[[m,l],[s],[h]]
Move’ = ’m12
NextState’ = ’[[l],[m],[s,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]]]
Move’ = ’m31
NextState’ = ’[[s,l],[m],[h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]]]
```

```
Move’ = ’m31
NextState’ = ’[[h,s,l],[m],[]]
Move’ = ’m32
NextState’ = ’[[s,l],[h,m],[]]
Move’ = ’m21
NextState’ = ’[[m,s,l],[],[h]]
Move’ = ’m23
NextState’ = ’[[s,l],[],[m,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]]]
Move’ = ’m31
NextState’ = ’[[m,s,l],[],[h]]
Move’ = ’m32
NextState’ = ’[[s,l],[m],[h]]
Move’ = ’m12
NextState’ = ’[[l],[s],[m,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]]]
Move’ = ’m31
NextState’ = ’[[m,l],[s],[h]]
```

```
Move' = 'm32
NextState' = '[[l],[m,s],[h]]
Move' = 'm21
NextState' = '[[s,l],[],[m,h]]
Move' = 'm23
NextState' = '[[l],[],[s,m,h]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]]]
Move' = 'm31
NextState' = '[[s,l],[],[m,h]]
Move' = 'm32
NextState' = '[[l],[s],[m,h]]
Move' = 'm12
NextState' = '[[],[l],[s,m,h]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]]]
Move' = 'm31
NextState' = '[[s],[l],[m,h]]
```

```
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]]]
Move’ = ’m31
NextState’ = ’[[m,s],[l],[h]]
Move’ = ’m32
NextState’ = ’[[s],[m,l],[h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]]]
Move’ = ’m31
NextState’ = ’[[h,s],[m,l],[]]
Move’ = ’m32
NextState’ = ’[[s],[h,m,l],[]]
Move’ = ’m21
NextState’ = ’[[m,s],[l],[h]]
Move’ = ’m23
NextState’ = ’[[s],[l],[m,h]]
```

```
Move’ = ’m12
NextState’ = ’[[],[s,m,l],[h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]]]
Move’ = ’m31
NextState’ = ’[[h],[s,m,l],[]]
Move’ = ’m32
NextState’ = ’[[],[h,s,m,l],[]]
Move’ = ’m21
NextState’ = ’[[s],[m,l],[h]]
Move’ = ’m23
NextState’ = ’[[],[m,l],[s,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]]]
Move’ = ’m31
NextState’ = ’[[s],[m,l],[h]]
```

```
Move’ = ’m32
NextState’ = ’[[],[s,m,l],[h]]
Move’ = ’m21
NextState’ = ’[[m],[l],[s,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]]]
Move’ = ’m31
NextState’ = ’[[s,m],[l],[h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]],[[s,m],[l],[h]]]
Move’ = ’m31
NextState’ = ’[[h,s,m],[l],[]]
Move’ = ’m32
NextState’ = ’[[s,m],[h,l],[]]
```

```
Move’ = ’m21
NextState’ = ’[[l,s,m],[],[h]]
Move’ = ’m23
NextState’ = ’[[s,m],[],[l,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]],[[s,m],[l],[h]],[[s,m],[],[l,h
]]]
Move’ = ’m31
NextState’ = ’[[l,s,m],[],[h]]
Move’ = ’m32
NextState’ = ’[[s,m],[l],[h]]
Move’ = ’m12
NextState’ = ’[[m],[s],[l,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]],[[s,m],[l],[h]],[[s,m],[],[l,h
]],[[m],[s],[l,h]]]
Move’ = ’m31
NextState’ = ’[[l,m],[s],[h]]
Move’ = ’m32
NextState’ = ’[[m],[l,s],[h]]
Move’ = ’m21
NextState’ = ’[[s,m],[],[l,h]]
Move’ = ’m23
NextState’ = ’[[m],[],[s,l,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]],[[s,m],[l],[h]],[[s,m],[],[l,h
]],[[m],[s],[l,h]],[[m],[],[s,l,h]]]
Move’ = ’m31
NextState’ = ’[[s,m],[],[l,h]]
Move’ = ’m32
NextState’ = ’[[m],[s],[l,h]]
Move’ = ’m12
NextState’ = ’[[],[m],[s,l,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]],[[s,m],[l],[h]],[[s,m],[],[l,h
]],[[m],[s],[l,h]],[[m],[],[s,l,h]],[[],[m],[s,l,h]]]
Move' = 'm31
NextState' = '[[s],[m],[l,h]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]],[[s,m],[l],[h]],[[s,m],[],[l,h
]],[[m],[s],[l,h]],[[m],[],[s,l,h]],[[],[m],[s,l,h]],[[s],[m],
[l,h]]]
Move' = 'm31
NextState' = '[[l,s],[m],[h]]
Move' = 'm32
NextState' = '[[s],[l,m],[h]]
Move' = 'm21
NextState' = '[[m,s],[],[l,h]]
Move' = 'm23
NextState' = '[[s],[],[m,l,h]]
PathSoFar' = '[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]],[[s,m],[l],[h]],[[s,m],[],[l,h
]],[[m],[s],[l,h]],[[m],[],[s,l,h]],[[],[m],[s,l,h]],[[s],[m],
[l,h]],[[s],[],[m,l,h]]]
Move’ = ’m31
NextState’ = ’[[m,s],[],[l,h]]
Move’ = ’m32
NextState’ = ’[[s],[m],[l,h]]
Move’ = ’m12
NextState’ = ’[[],[s],[m,l,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]],[[s,m],[l],[h]],[[s,m],[],[l,h
]],[[m],[s],[l,h]],[[m],[],[s,l,h]],[[],[m],[s,l,h]],[[s],[m],
[l,h]],[[s],[],[m,l,h]],[[],[s],[m,l,h]]]
Move’ = ’m31
NextState’ = ’[[m],[s],[l,h]]
Move’ = ’m32
NextState’ = ’[[],[m,s],[l,h]]
Move’ = ’m21
NextState’ = ’[[s],[],[m,l,h]]
```

```
Move’ = ’m23
NextState’ = ’[[],[],[s,m,l,h]]
PathSoFar’ = ’[[[s,m,l,h],[],[]],[[m,l,h],[s],[]],[[m,l,h],[],[s]],
[[l,h],[m],[s]],[[s,l,h],[m],[]],[[s
,l,h],[],[m]],[[l,h],[s],[m]],[[l,h],[],[s,m]],[[h],[l],[s,m]],
[[s,h],[l],[m]],[[s,h],[m,l],[]],[[h],[s
,m,l],[]],[[h],[m,l],[s]],[[m,h],[l],[s]],[[s,m,h],[l],[]],[[s,m,h],
[],[l]],[[m,h],[s],[l]],[[m,h],[],[
s,l]],[[h],[m],[s,l]],[[s,h],[m],[l]],[[s,h],[],[m,l]],[[h],[s],
[m,l]],[[h],[],[s,m,l]],[[],[h],[s,m,l]
],[[s],[h],[m,l]],[[s],[m,h],[l]],[[],[s,m,h],[l]],[[l],[s,m,h],[]],
[[s,l],[m,h],[]],[[s,l],[h],[m]],[[
l],[s,h],[m]],[[m,l],[s,h],[]],[[s,m,l],[h],[]],[[s,m,l],[],[h]],
[[m,l],[s],[h]],[[m,l],[],[s,h]],[[l],
[m],[s,h]],[[s,l],[m],[h]],[[s,l],[],[m,h]],[[l],[s],[m,h]],[[l],[],
[s,m,h]],[[],[l],[s,m,h]],[[s],[l],
[m,h]],[[s],[m,l],[h]],[[],[s,m,l],[h]],[[],[m,l],[s,h]],[[m],[l],
[s,h]],[[s,m],[l],[h]],[[s,m],[],[l,h
]],[[m],[s],[l,h]],[[m],[],[s,l,h]],[[],[m],[s,l,h]],[[s],[m],
[l,h]],[[s],[],[m,l,h]],[[],[s],[m,l,h]],
[[],[],[s,m,l,h]]]
SolutionSoFar’ =
’[m12,m23,m12,m31,m23,m12,m23,m12,m31,m32,m12,m23,m21,m31,m23,m12,m2
3,m12,m31,m23,m12,
m23,m12,m31,m32,m12,m31,m21,m23,m12,m31,m21,m23,m12,m23,m12,m31,m23,
m12,m23,m12,m31,m32,m12,m23,m21,m31
,m23,m12,m23,m12,m31,m23,m12,m23]

Solution ...

Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 3 to tower 2
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 2 to tower 1
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 3 to tower 2
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 3 to tower 2
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 2 to tower 1
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 3 to tower 1
Transfer a disk from tower 2 to tower 3
Transfer a disk from tower 1 to tower 2
Transfer a disk from tower 2 to tower 3

true .

?-
```

Do your best to answer the following questions:

1. What was the length of your program’s solution to the four disk problem?

55

2. What is the length of the shortest solution to the four disk problem?

16