## Learning Abstract:

## Task 1:

Contemplate the nature of the problem, see specification on web page for details.

## Task 2:

Copy and paste source code and check to ensure validity and that it initially compiles, see specification on web page for details, full code posted later this document.

## Task 3: One Move Predicate and a Unit Test

m12([Tower1Before,Tower2Before,Tower3],[Tower1After,Tower2After,Tower3]) :-

    Tower1Before = [H|T],

    Tower1After = T,

    Tower2Before = L,

    Tower2After = [H|L].

test\_\_m12 :-

    write('Testing: move\_m12\n'),

    TowersBefore = [[t,s,m,l,h],[],[]],

    trace('','TowersBefore',TowersBefore),

    m12(TowersBefore,TowersAfter),

    trace('','TowersAfter',TowersAfter).

Text

Description automatically generated

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

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 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).

A screenshot of a computer

Description automatically generated with medium confidence