**Aim:** To implement Monkey Banana Problem in Prolog.

**Theory:**

A monkey is in a room. Suspended from the ceiling is a bunch of bananas, beyond the monkey's reach. However, in the room there are also a chair and a stick. The ceiling is just the right height so that a monkey standing on a chair could knock the bananas down with the stick. The monkey knows how to move around, carry other things around, reach for the bananas, and wave a stick in the air.

**Code:**

state(mx, my, bx, h).

do(state(atdoor, onfloor, atwindow, no),

walk(atdoor, atwindow),

state(atwindow, onfloor, atwindow, no)).

do(state(atwindow, onfloor, atwindow, no),

push(atwindow, middle),

state(middle, onfloor, middle, no)).

do(state(middle, onfloor, middle, no),

climb,

state(middle, onbox, middle, no)).

do(state(middle, onbox, middle, no),

grab,

state(middle, onbox, middle, yes)).

canget(state(\_, \_, \_, yes)).

canget(State1):-

do(State1, Action, State2),

canget(State2).

canget(state(\_, \_, \_, yes),[]).

canget(State1, Plan) :-

do(State1, Action, State2),

canget(State2, PartialPlan),

add(Action, PartialPlan, Plan).

add(X,L,[X|L]).

**Output:**

?- canget(state(atdoor, onfloor, atwindow, hasnot), Plan).

Plan = [walk(atdoor, atwindow), push(atwindow, middle), climb, grasp]

Yes

?- canget(state(atwindow, onbox, atwindow, hasnot), Plan ).

No

?- canget(state(Monkey, onfloor, atwindow, hasnot), Plan).

Monkey = atwindow

Plan = [push(atwindow, middle), climb, grasp]

Yes

**Conclusion:**

Hence we have implemented monkey banana problem in prolog.