Name: Brijesh Rameshbhai Rohit

Admission number: U19CS009

ARTIFICIAL INTELLIGENCE

ASSIGNMENT - 06

Monkey is on the floor, at the door. A block is on the floor, at the window. Bananas are hanging from the roof in the middle of the room.

Task is to get monkey bananas.

CODE=>

```
%U19CS009
%BRIJESH ROHIT

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Description %

%state(Pm1,Pm2,Pb,flag).
%Pm1 = Position of Monkey in the room (at door, at window, at middle).
%Pm2 = Position of Monkey in the room (on box, on floor).
%Pb = Position of Box in the room.
%flag = Whether monkey has the banana or not (has, hasnot).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%move(initial state, action to perform, next state).
move(
   state(middle,box,middle,hasnot),
   grab,
   state(middle,box,middle,has)
).

move(
   state(P,floor,P,H),
   climb,
   state(P,box,P,H)
).
```

```
move(
   state(P1,floor,P1,H),
   push(P1,P2),
   state(P2,floor,P2,H)
).

move(
   state(P1,floor,B,H),
   walk(P1,P2),
   state(P2,floor,B,H)
).

%predicate to display steps according to action M.
%action(action performed)

action(M):-
   M=walk(P1,P2),
   write("walk \t "),
   write(P1),
   write("    \t "),
   write(P2).

action(M):-
   M=push(P1,P2),
   write("push \t "),
   write(P1),
   write("    \t "),
   write(P2).

action(M):-
   M=grab,
   write("grab "),
   write("    -"),
   write("    \t "),
   write("-").

action(M):-
   M=climb,
```

```
   write("climb "),
   write("    -"),
   write("    \t "),
   write("-").

%predicate to display the Steps.
display([]).
display([H1|[]],[H2|[]]):-
   nl,
   write(H2),
   write(" \t"),
   action(H1).

display([H1|T1],[H2|T2]):-
   display(T1,T2),
   nl,
   write(H2),
   write(" \t"),
   action(H1).

%canget(State,List to append action predicate,List to append current
%state predicate).
canget(state(_,_,_,has),L,L1):-
   display(L,L1),
   nl,
   write("\nMonkey got the banana!"),
   nl.
canget(S,T,T1):-
   move(S,M,S1),
   canget(S1,[M|T],[S|T1]).

%start(State). predicate to start the program.
start(S1):-
   write("\nInitial State: "),
   write(S1),
   nl,nl,
   write("Current State\t\t\t\tAction\t Old Position\t New Position"),
   canget(S1,[],[]).
```

OUTPUT=>

1. Initial state

- Monkey at door
- Monkey on floor
- Box at window
- Monkey does not have the banana

```
?- ['/home/brijesh/Documents/ai/ai-assign06/u19cs009-ai-assign06-monkey.pl'].
true.

?- start(state(door,floor,window,hasnot)).

Initial State: state(door,floor,window,hasnot)

Current State                          Action   Old Position    New Position
state(door,floor,window,hasnot)        walk     door            window
state(window,floor,window,hasnot)      push     window          middle
state(middle,floor,middle,hasnot)      climb     -      -
state(middle,box,middle,hasnot)        grab     -       -

Monkey got the banana!
true .
```

2. Initial state

- Monkey at window
- Monkey on floor
- Box at window
- Monkey does not have the banana

```
?- start(state(window,floor,window,hasnot)).

Initial State: state(window,floor,window,hasnot)

Current State                          Action   Old Position    New Position
state(window,floor,window,hasnot)      push     window          middle
state(middle,floor,middle,hasnot)      climb     -      -
state(middle,box,middle,hasnot)        grab     -       -

Monkey got the banana!
true .
```

3. Initial state

- Monkey at middle
- Monkey on floor
- Box at window
- Monkey does not have the banana

```
?- start(state(middle,floor,window,hasnot)).

Initial State: state(middle,floor,window,hasnot)

Current State                          Action   Old Position    New Position
state(middle,floor,window,hasnot)      walk     middle          window
state(window,floor,window,hasnot)      push     window          middle
state(middle,floor,middle,hasnot)      climb     -      -
state(middle,box,middle,hasnot)        grab     -       -

Monkey got the banana!
true .
```