19.Write a Prolog program to implement Monkey Banana Problem.

Program:

```
move(state(middle,onbox,middle,hasnot),
   grasp,
   state(middle,onbox,middle,has)).
move(state(P,onfloor,P,H),
   climb,
   state(P,onbox,P,H)).
move(state(P1,onfloor,P1,H),
   drag(P1,P2),
   state(P2,onfloor,P2,H)).
move(state(P1,onfloor,B,H),
   walk(P1,P2),
   state(P2,onfloor,B,H)).
canget(state(_,_,_,has)).
canget(State1) :-
   move(State1,_,State2),
   canget(State2).
```

OUTPUT:

```
% c:/users/rohith kumar/onedrive/documents/prolog/money banana problem compiled 0.00 sec, 0 clauses
?- |    canget(state(atdoor, onfloor, atwindow, hasnot)).
true .

?- trace.
true.

[trace]  ?- canget(state(atdoor, onfloor, atwindow, hasnot)).
   Call: (10) canget(state(atdoor, onfloor, atwindow, hasnot)) ? creep
   Call: (11) move(state(atdoor, onfloor, atwindow, hasnot), _11306, _11246) ? creep
   Exit: (11) move(state(atdoor, onfloor, atwindow, hasnot), walk(atdoor, _12010), state(_12010, onfloor, atwindow, hasnot)) ? creep
   Call: (11) canget(state(_12010, onfloor, atwindow, hasnot)) ? creep
   Call: (12) move(state(_12010, onfloor, atwindow, hasnot), _13590, _13530) ? creep
   Exit: (12) move(state(atwindow, onfloor, atwindow, hasnot), climb, state(atwindow, onbox, atwindow, hasnot)) ? creep
   Call: (12) canget(state(atwindow, onbox, atwindow, hasnot)) ? creep
   Call: (13) move(state(atwindow, onbox, atwindow, hasnot), _15868, _15808) ? creep
   Fail: (13) move(state(atwindow, onbox, atwindow, hasnot), _16626, _15808) ? creep
   Fail: (12) canget(state(atwindow, onbox, atwindow, hasnot)) ? creep
   Redo: (12) move(state(_12010, onfloor, atwindow, hasnot), _18134, _13530) ? creep
   Exit: (12) move(state(atwindow, onfloor, atwindow, hasnot), drag(atwindow, _18838), state(_18838, onfloor, _18838, hasnot)) ? creep
   Call: (12) canget(state(_18838, onfloor, _18838, hasnot)) ? creep
   Call: (13) move(state(_18838, onfloor, _18838, hasnot), _20418, _20358) ? creep
   Exit: (13) move(state(_18838, onfloor, _18838, hasnot), climb, state(_18838, onbox, _18838, hasnot)) ? creep
   Call: (13) canget(state(_18838, onbox, _18838, hasnot)) ? creep
   Call: (14) move(state(_18838, onbox, _18838, hasnot), _22696, _22636) ? creep
   Exit: (14) move(state(middle, onbox, middle, hasnot), grasp, state(middle, onbox, middle, has)) ? creep
   Call: (14) canget(state(middle, onbox, middle, has)) ? creep
   Exit: (14) canget(state(middle, onbox, middle, has)) ? creep
   Exit: (13) canget(state(middle, onbox, middle, hasnot)) ? creep
   Exit: (12) canget(state(middle, onfloor, middle, hasnot)) ? creep
   Exit: (11) canget(state(atwindow, onfloor, atwindow, hasnot)) ? creep
   Exit: (10) canget(state(atdoor, onfloor, atwindow, hasnot)) ? creep
true .
```