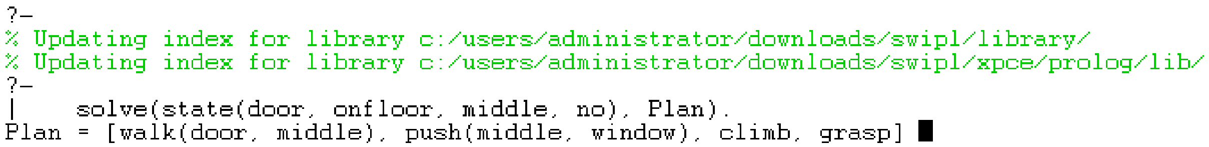
Write a Prolog program to implement Monkey Banana Problem

```
% Monkey and Banana Problem

% State: state(MonkeyPos, MonkeyOnBox, BoxPos, HasBanana)

goal(state(_, _, _, has)).

% Actions

move(state(door, onfloor, middle, no), walk(door, middle),

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

move(state(middle, onfloor, middle, no), push(middle, window),

    state(window, onfloor, window, no)).

move(state(window, onfloor, window, no), climb,

    state(window, onbox, window, no)).

move(state(window, onbox, window, no), grasp,

    state(window, onbox, window, has)).


% Solver

solve(State, []) :- goal(State).

solve(State, [Act|Plan]) :-

   move(State, Act, NewState),

   solve(NewState, Plan).
```



```
?-
% Updating index for library c:/users/administrator/downloads/swipl/library/
% Updating index for library c:/users/administrator/downloads/swipl/xpce/prolog/lib/
?-
|    solve(state(door, onfloor, middle, no), Plan).
Plan = [walk(door, middle), push(middle, window), climb, grasp]
```