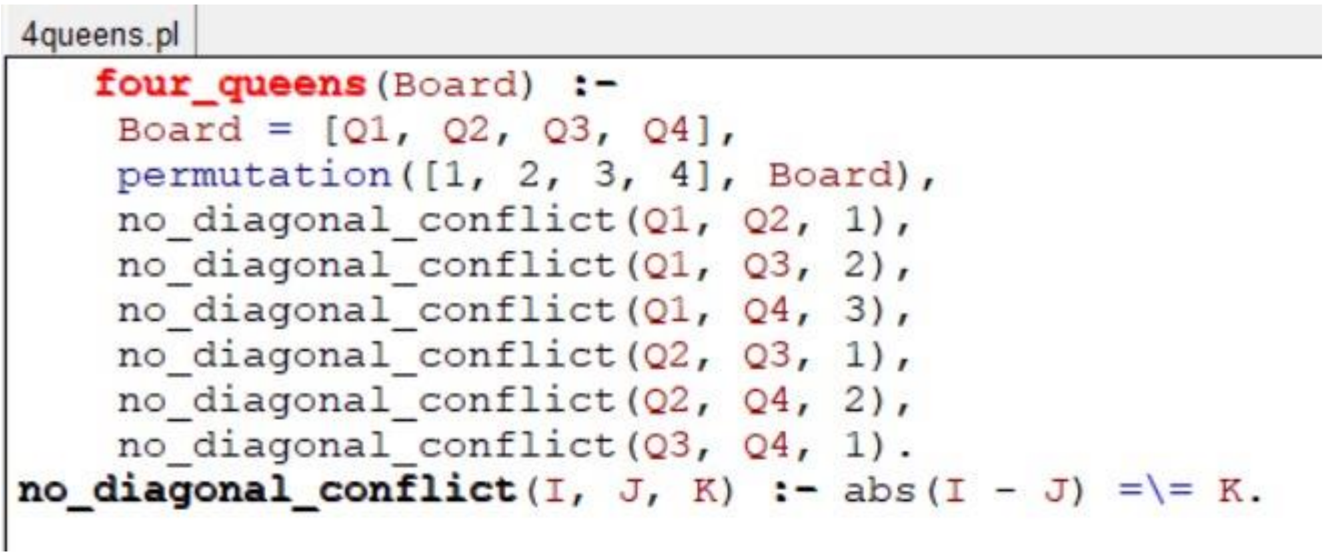
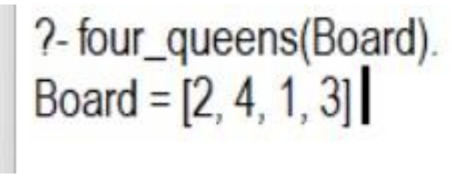
Code :



4queens.pl

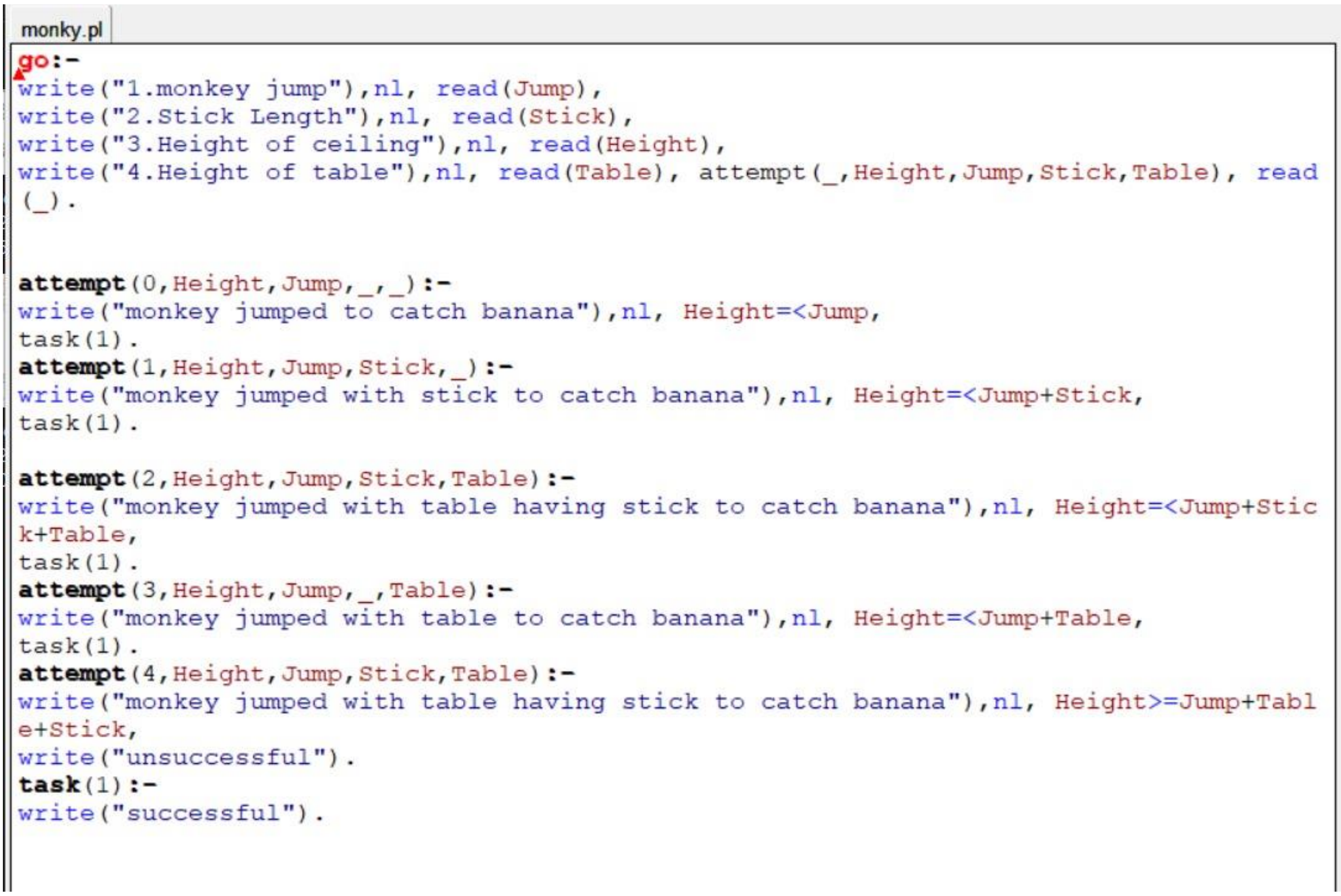
```
    four_queens(Board) :-
     Board = [Q1, Q2, Q3, Q4],
     permutation([1, 2, 3, 4], Board),
     no_diagonal_conflict(Q1, Q2, 1),
     no_diagonal_conflict(Q1, Q3, 2),
     no_diagonal_conflict(Q1, Q4, 3),
     no_diagonal_conflict(Q2, Q3, 1),
     no_diagonal_conflict(Q2, Q4, 2),
     no_diagonal_conflict(Q3, Q4, 1).
no_diagonal_conflict(I, J, K) :- abs(I - J) =\= K.
```

Output :



```
?- four_queens(Board).
Board = [2, 4, 1, 3]
```

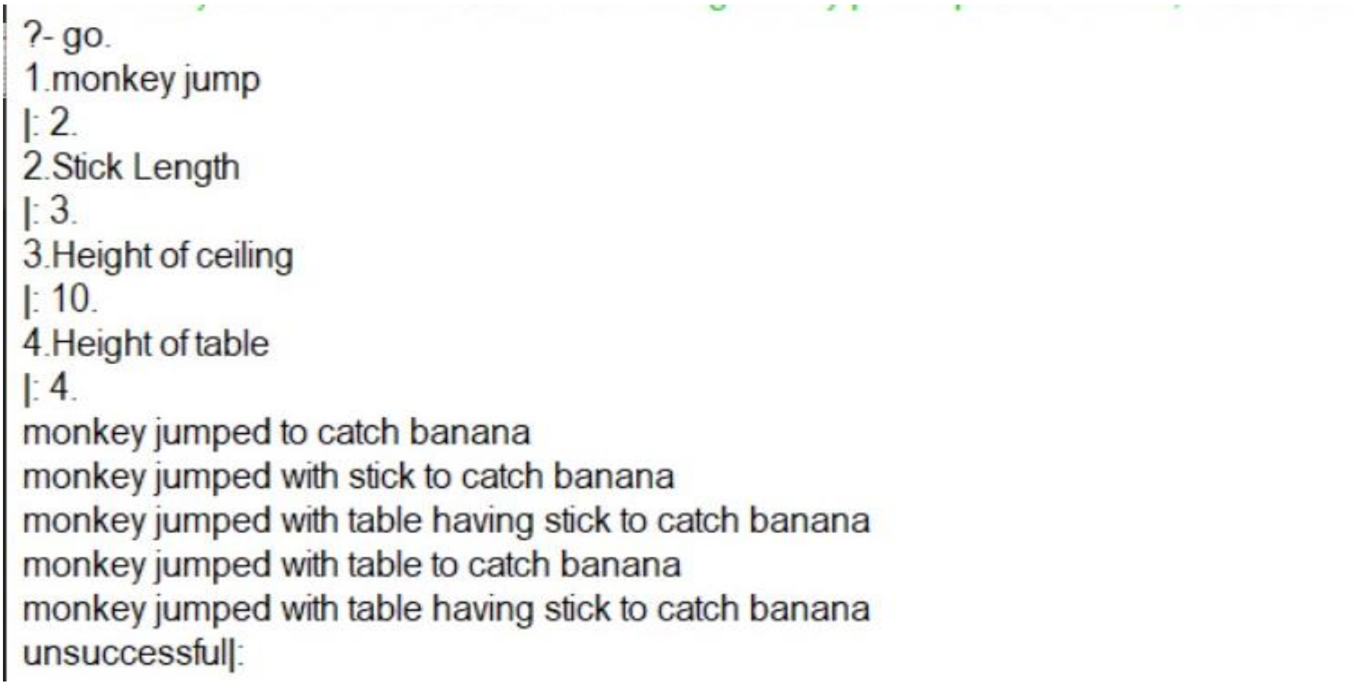
Code :



monky.pl

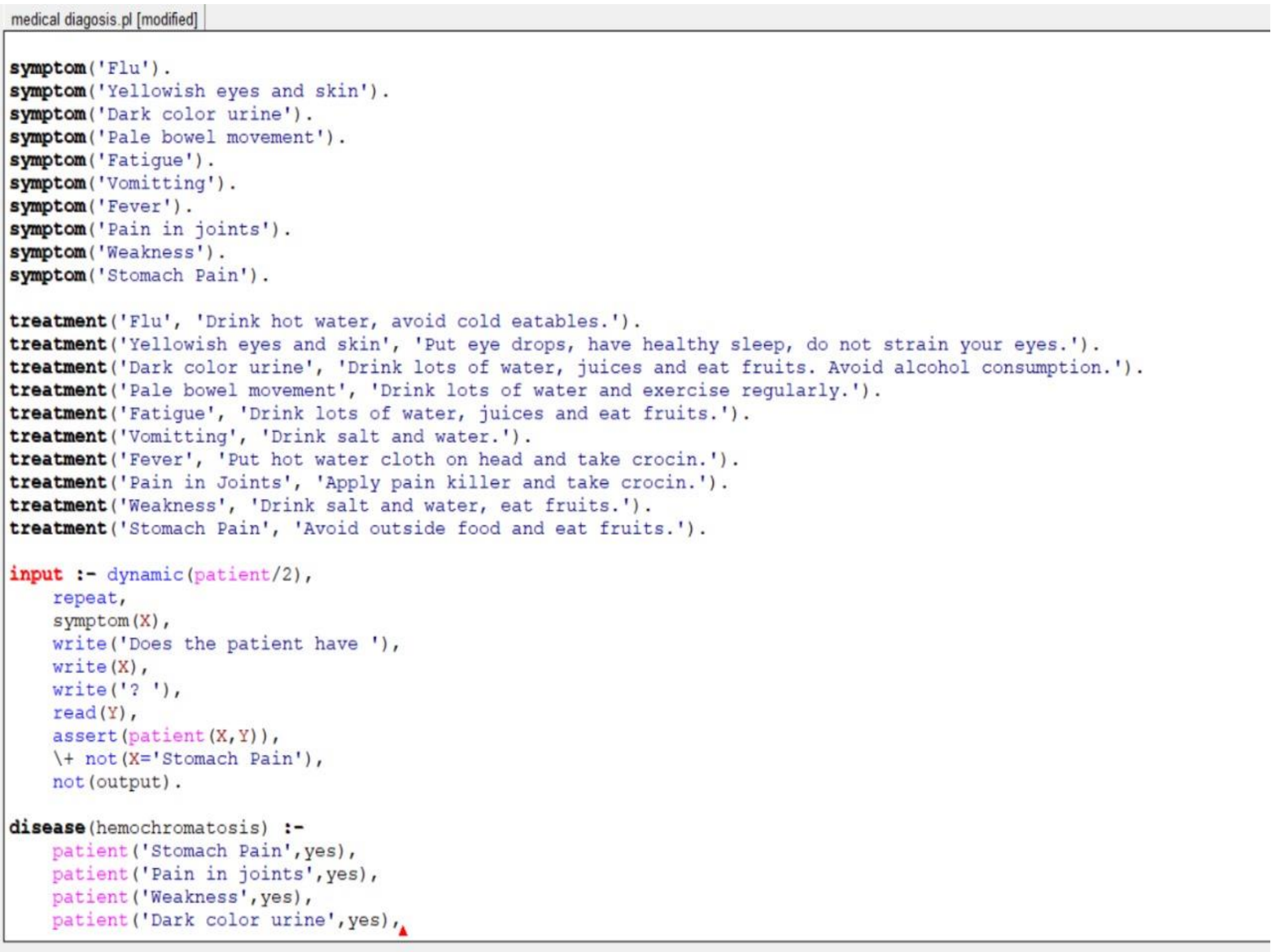
```
go:-
write("1.monkey jump"),nl, read(Jump),
write("2.Stick Length"),nl, read(Stick),
write("3.Height of ceiling"),nl, read(Height),
write("4.Height of table"),nl, read(Table), attempt(_,Height,Jump,Stick,Table), read
(_).


attempt(0,Height,Jump,_,_):-
write("monkey jumped to catch banana"),nl, Height=<Jump,
task(1).
attempt(1,Height,Jump,Stick,_):-
write("monkey jumped with stick to catch banana"),nl, Height=<Jump+Stick,
task(1).

attempt(2,Height,Jump,Stick,Table):-
write("monkey jumped with table having stick to catch banana"),nl, Height=<Jump+Stic
k+Table,
task(1).
attempt(3,Height,Jump,_,Table):-
write("monkey jumped with table to catch banana"),nl, Height=<Jump+Table,
task(1).
attempt(4,Height,Jump,Stick,Table):-
write("monkey jumped with table having stick to catch banana"),nl, Height>=Jump+Tabl
e+Stick,
write("unsuccessful").
task(1):-
write("successful").
```

Output :

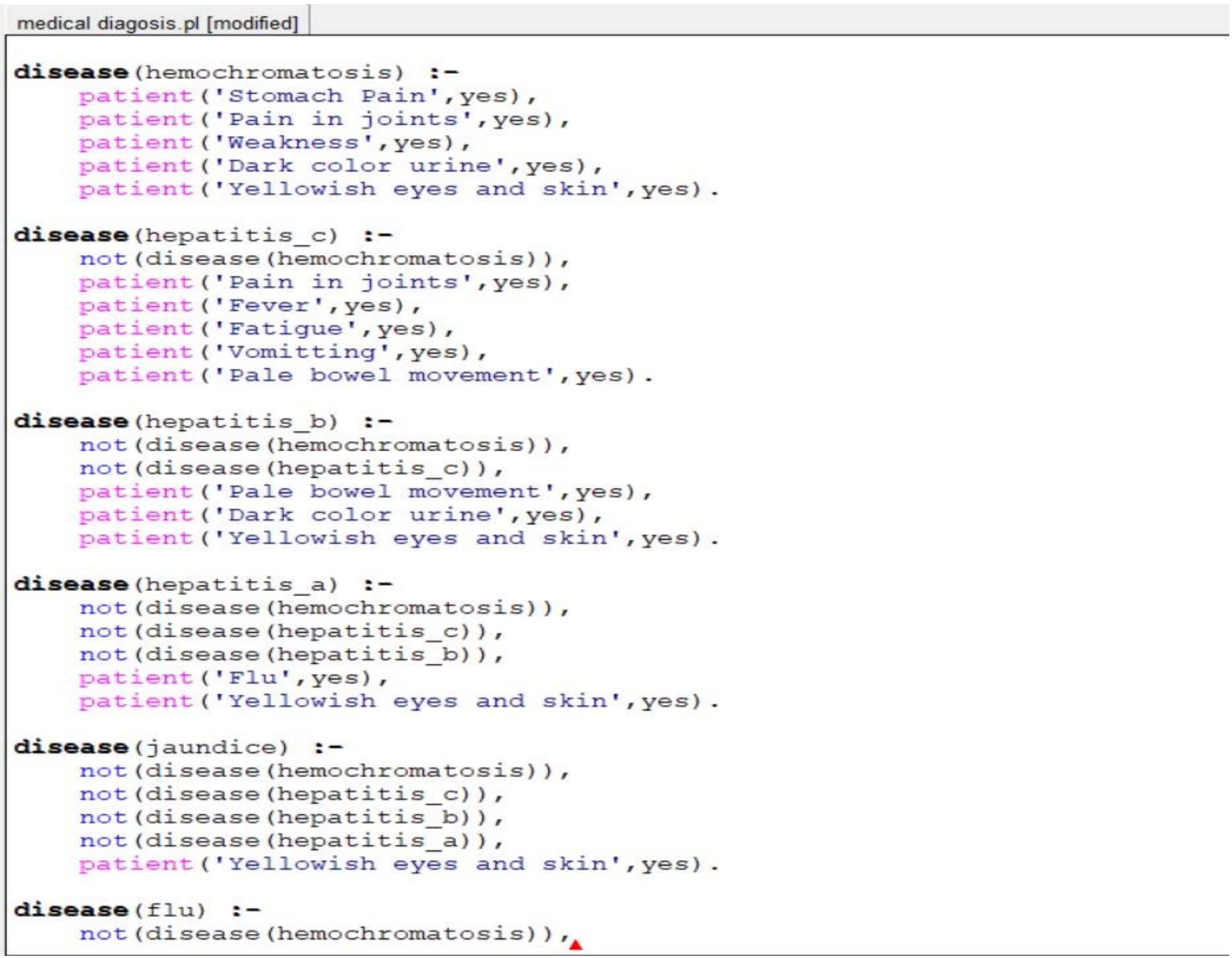


```
?- go.
1.monkey jump
|: 2.
2.Stick Length
|: 3.
3.Height of ceiling
|: 10.
4.Height of table
|: 4.
monkey jumped to catch banana
monkey jumped with stick to catch banana
monkey jumped with table having stick to catch banana
monkey jumped with table to catch banana
monkey jumped with table having stick to catch banana
unsuccessful|:
```

Code :

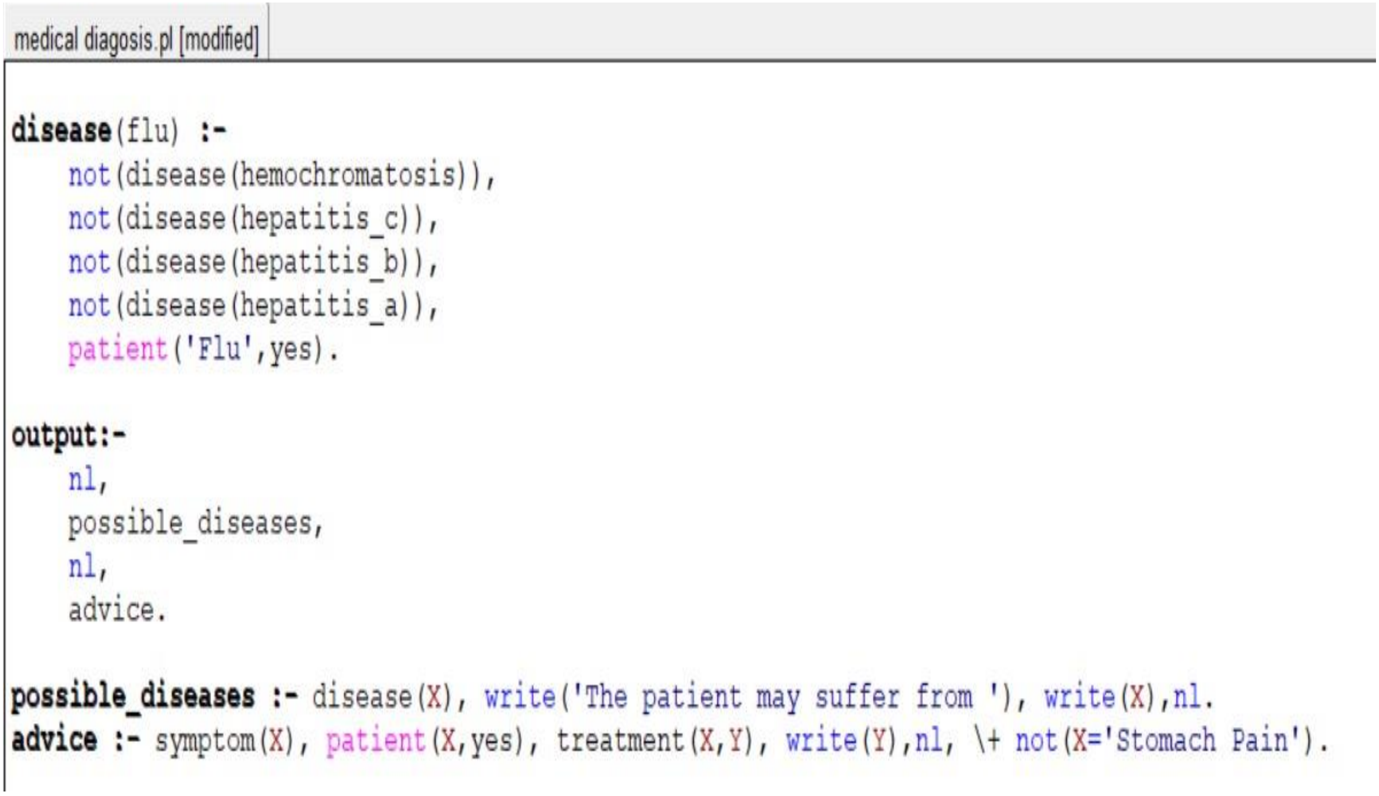


```
symptom('Flu').
symptom('Yellowish eyes and skin').
symptom('Dark color urine').
symptom('Pale bowel movement').
symptom('Fatigue').
symptom('Vomitting').
symptom('Fever').
symptom('Pain in joints').
symptom('Weakness').
symptom('Stomach Pain').

treatment('Flu', 'Drink hot water, avoid cold eatables.').
treatment('Yellowish eyes and skin', 'Put eye drops, have healthy sleep, do not strain your eyes.').
treatment('Dark color urine', 'Drink lots of water, juices and eat fruits. Avoid alcohol consumption.').
treatment('Pale bowel movement', 'Drink lots of water and exercise regularly.').
treatment('Fatigue', 'Drink lots of water, juices and eat fruits.').
treatment('Vomitting', 'Drink salt and water.').
treatment('Fever', 'Put hot water cloth on head and take crocin.').
treatment('Pain in Joints', 'Apply pain killer and take crocin.').
treatment('Weakness', 'Drink salt and water, eat fruits.').
treatment('Stomach Pain', 'Avoid outside food and eat fruits.').

input :- dynamic(patient/2),
    repeat,
    symptom(X),
    write('Does the patient have '),
    write(X),
    write('? '),
    read(Y),
    assert(patient(X,Y)),
    \+ not(X='Stomach Pain'),
    not(output).

disease(hemochromatosis) :-
    patient('Stomach Pain',yes),
    patient('Pain in joints',yes),
    patient('Weakness',yes),
    patient('Dark color urine',yes),
```



medical diagosis.pl [modified]

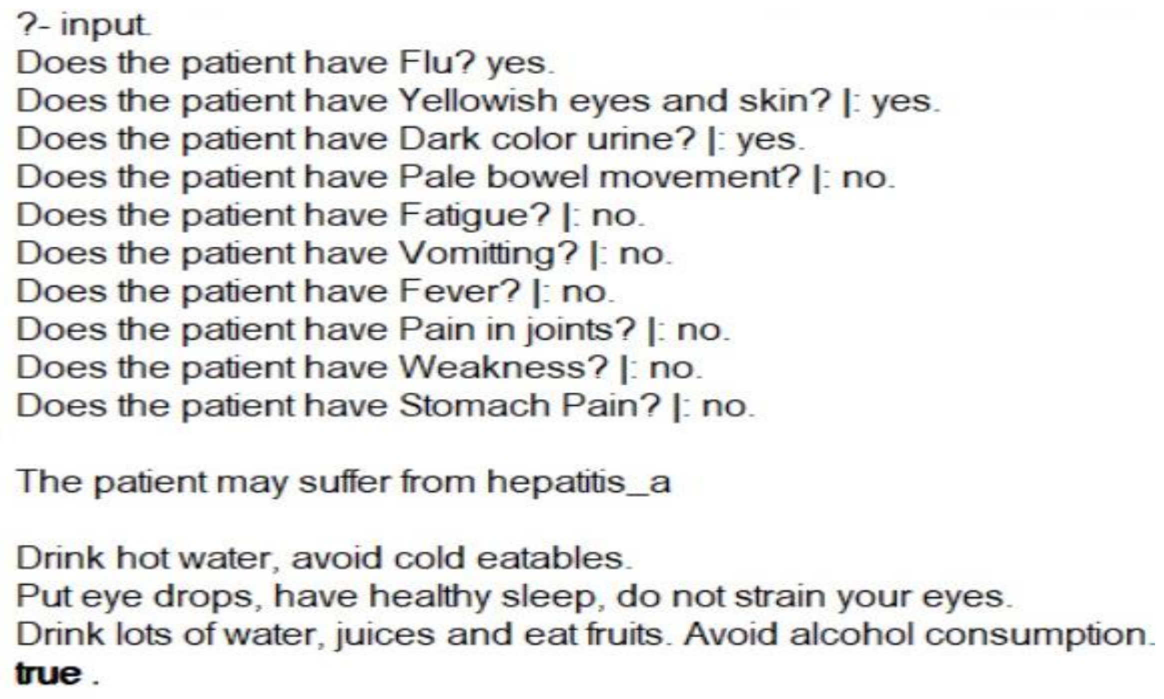
```
disease(hemochromatosis) :-
     patient('Stomach Pain',yes),
     patient('Pain in joints',yes),
     patient('Weakness',yes),
     patient('Dark color urine',yes),
     patient('Yellowish eyes and skin',yes).

disease(hepatitis_c) :-
     not(disease(hemochromatosis)),
     patient('Pain in joints',yes),
     patient('Fever',yes),
     patient('Fatigue',yes),
     patient('Vomitting',yes),
     patient('Pale bowel movement',yes).

disease(hepatitis_b) :-
     not(disease(hemochromatosis)),
     not(disease(hepatitis_c)),
     patient('Pale bowel movement',yes),
     patient('Dark color urine',yes),
     patient('Yellowish eyes and skin',yes).

disease(hepatitis_a) :-
     not(disease(hemochromatosis)),
     not(disease(hepatitis_c)),
     not(disease(hepatitis_b)),
     patient('Flu',yes),
     patient('Yellowish eyes and skin',yes).

disease(jaundice) :-
     not(disease(hemochromatosis)),
     not(disease(hepatitis_c)),
     not(disease(hepatitis_b)),
     not(disease(hepatitis_a)),
     patient('Yellowish eyes and skin',yes).

disease(flu) :-
     not(disease(hemochromatosis)),
```



medical diagosis.pl [modified]

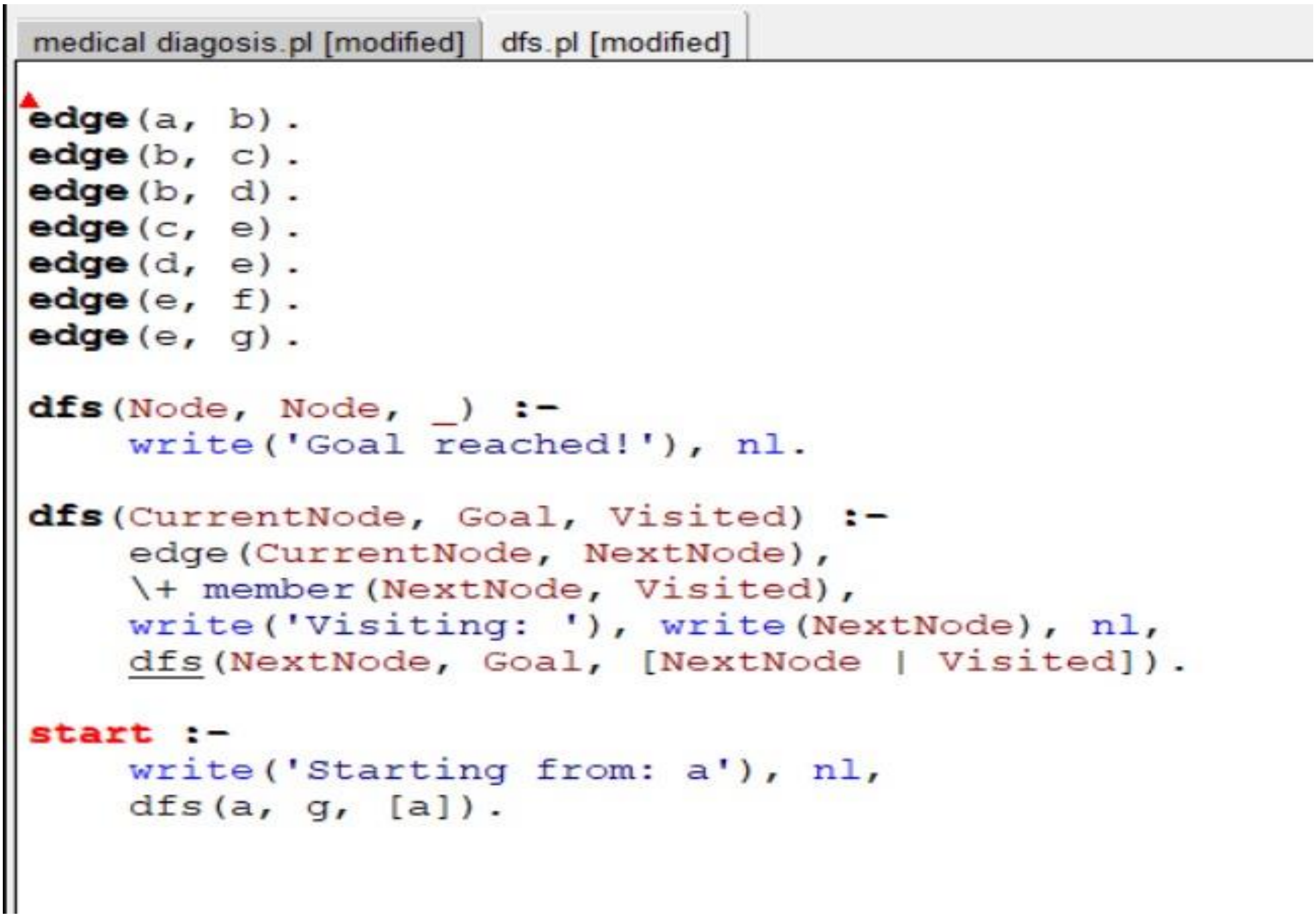
```
disease(flu) :-
    not(disease(hemochromatosis)),
    not(disease(hepatitis_c)),
    not(disease(hepatitis_b)),
    not(disease(hepatitis_a)),
    patient('Flu',yes).

output:-
    nl,
    possible_diseases,
    nl,
    advice.

possible_diseases :- disease(X), write('The patient may suffer from '), write(X),nl.
advice :- symptom(X), patient(X,yes), treatment(X,Y), write(Y),nl, \+ not(X='Stomach Pain').
```

Output :



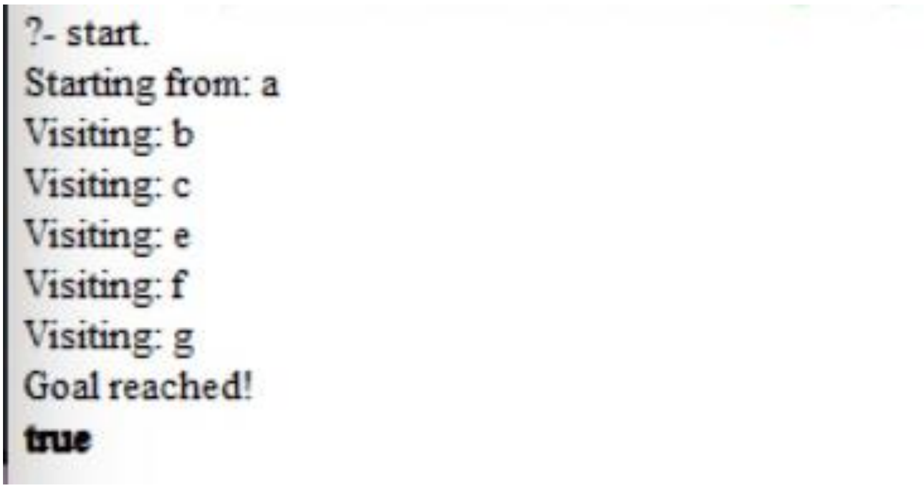
```
?- input.
Does the patient have Flu? yes.
Does the patient have Yellowish eyes and skin? |: yes.
Does the patient have Dark color urine? |: yes.
Does the patient have Pale bowel movement? |: no.
Does the patient have Fatigue? |: no.
Does the patient have Vomitting? |: no.
Does the patient have Fever? |: no.
Does the patient have Pain in joints? |: no.
Does the patient have Weakness? |: no.
Does the patient have Stomach Pain? |: no.

The patient may suffer from hepatitis_a

Drink hot water, avoid cold eatables.
Put eye drops, have healthy sleep, do not strain your eyes.
Drink lots of water, juices and eat fruits. Avoid alcohol consumption.
true .
```

Code :



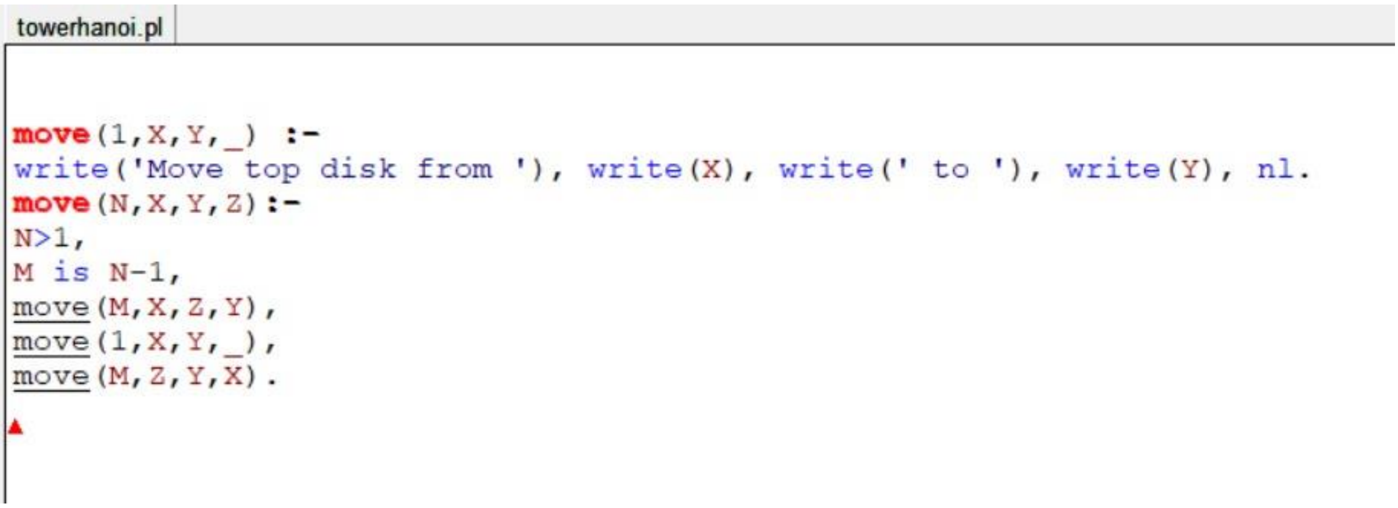
```
edge(a, b).
edge(b, c).
edge(b, d).
edge(c, e).
edge(d, e).
edge(e, f).
edge(e, g).

dfs(Node, Node, _) :-
    write('Goal reached!'), nl.

dfs(CurrentNode, Goal, Visited) :-
    edge(CurrentNode, NextNode),
    \+ member(NextNode, Visited),
    write('Visiting: '), write(NextNode), nl,
    dfs(NextNode, Goal, [NextNode | Visited]).

start :-
    write('Starting from: a'), nl,
    dfs(a, g, [a]).
```

Output :



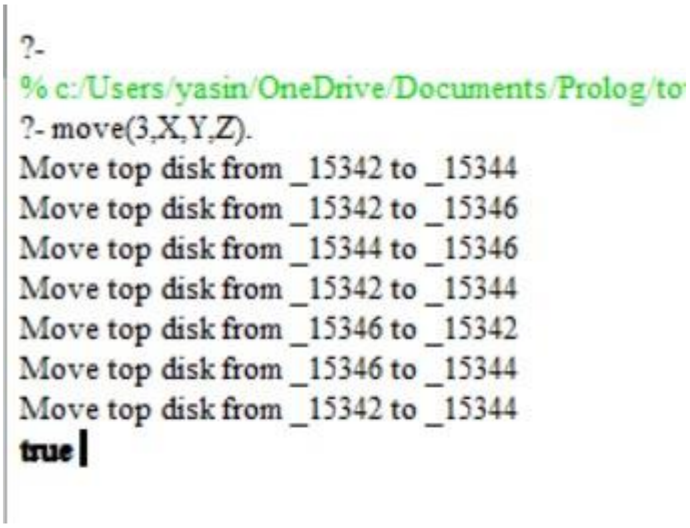
```
?- start.
Starting from: a
Visiting: b
Visiting: c
Visiting: e
Visiting: f
Visiting: g
Goal reached!
true
```

Code :



```
move(1,X,Y,_) :-
write('Move top disk from '), write(X), write(' to '), write(Y), nl.
move(N,X,Y,Z):-
N>1,
M is N-1,
move(M,X,Z,Y),
move(1,X,Y,_),
move(M,Z,Y,X).
```

Output :



```
?-
% c:/Users/yasin/OneDrive/Documents/Prolog/to
?- move(3,X,Y,Z).
Move top disk from _15342 to _15344
Move top disk from _15342 to _15346
Move top disk from _15344 to _15346
Move top disk from _15342 to _15344
Move top disk from _15346 to _15342
Move top disk from _15346 to _15344
Move top disk from _15342 to _15344
true
```