29.Write a Prolog Program for backward Chaining. Incorporate required queries.

% Declare symptom as dynamic to allow user interaction or facts

:- dynamic(symptom/1).

% Rules: disease is diagnosed if all required symptoms are present

disease(flu) :-

symptom(fever),

symptom(cough),

symptom(body\_ache).

disease(cold) :-

symptom(cough),

symptom(sneezing),

symptom(runny\_nose).

disease(covid19) :-

symptom(fever),

symptom(cough),

symptom(loss\_of\_taste),

symptom(breathing\_difficulty).

disease(malaria) :-

symptom(fever),

symptom(chills),

symptom(sweating).

% Ask user if a symptom exists

ask(Symptom) :-

format('Do you have ~w? (yes/no): ', [Symptom]),

read(Response),

( (Response == yes ; Response == y) ->

assertz(symptom(Symptom)) ;

fail ).

% Check if a symptom is known, else ask the user

check(Symptom) :-

symptom(Symptom), !.

check(Symptom) :-

ask(Symptom).

% Redefine each rule using 'check' instead of 'symptom' to support user input

diagnose(flu) :-

check(fever),

check(cough),

check(body\_ache).

diagnose(cold) :-

check(cough),

check(sneezing),

check(runny\_nose).

diagnose(covid19) :-

check(fever),

check(cough),

check(loss\_of\_taste),

check(breathing\_difficulty).

diagnose(malaria) :-

check(fever),

check(chills),

check(sweating).

% Main predicate

start :-

write('Backward Chaining Medical Diagnosis System'), nl,

diagnose(Disease),

format('Based on your symptoms, you may have: ~w~n', [Disease]), !.

start :-

write('Sorry, we could not diagnose your condition based on given symptoms.'), nl.

OUTPUT:

