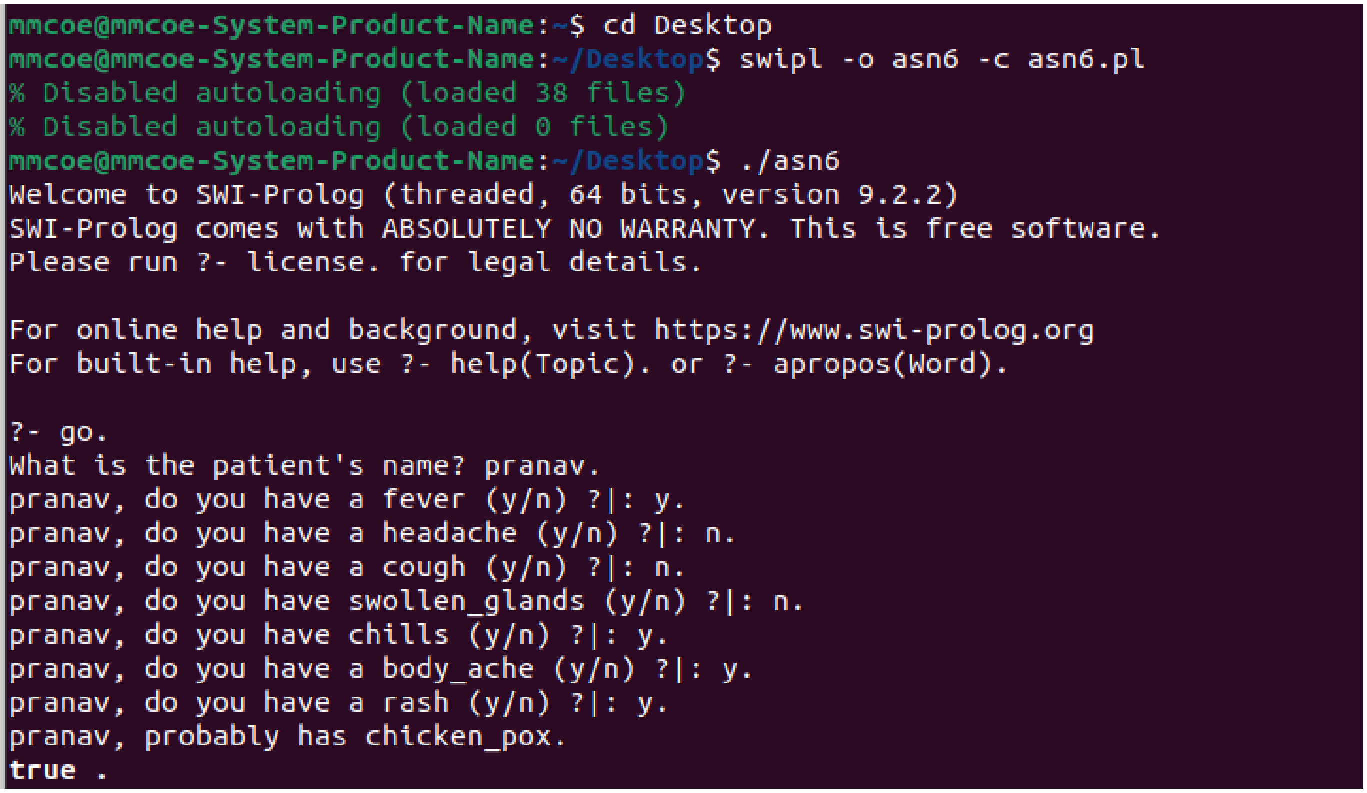
```
go :-
        write('What is the patient"s name? '),
        read(Patient),
        hypothesis(Patient, Disease),
        write_list([Patient,', probably has ',Disease,'.']), nl.

go :-
        write('Sorry, I don't seem to be able to'), nl,
        write('diagnose the disease.'), nl.

symptom(Patient, fever) :-
        verify(Patient, " have a fever (y/n) ?").
symptom(Patient, rash) :-
        verify(Patient, " have a rash (y/n) ?").
symptom(Patient, headache) :-
        verify(Patient, " have a headache (y/n) ?").
symptom(Patient, runny_nose) :-
        verify(Patient, " have a runny_nose (y/n) ?").
symptom(Patient, conjunctivitis) :-
        verify(Patient, " have conjunctivitis (y/n) ?").
symptom(Patient, cough) :-
        verify(Patient, " have a cough (y/n) ?").
symptom(Patient, body_ache) :-
        verify(Patient, " have a body_ache (y/n) ?").
symptom(Patient, chills) :-
        verify(Patient, " have chills (y/n) ?").
symptom(Patient, sore_throat) :-
        verify(Patient, " have a sore_throat (y/n) ?").
symptom(Patient, sneezing) :-
        verify(Patient, " have sneezing (y/n) ?").
symptom(Patient, swollen_glands) :-
        verify(Patient, " have swollen_glands (y/n) ?").

ask(Patient, Question) :-
        write(Patient), write(', do you'), write(Question),
        read(N),
        ( (N == yes ; N == y)
        ->
        assert(yes(Question)) ;
        assert(no(Question)), fail).

:- dynamic yes/1, no/1.

verify(P, S) :-
```

```
  (yes(S) -> true ;
        (no(S) -> fail ;
        ask(P, S))).

undo :- retract(yes(_)),fail.
undo :- retract(no(_)),fail.

hypothesis(Patient, german_measles) :-
        symptom(Patient, fever),
        symptom(Patient, headache),
        symptom(Patient, runny_nose),
        symptom(Patient, rash).

hypothesis(Patient, common_cold) :-
        symptom(Patient, headache),
        symptom(Patient, sneezing),
        symptom(Patient, sore_throat),
        symptom(Patient, runny_nose),
        symptom(Patient, chills).

hypothesis(Patient, measles) :-
        symptom(Patient, cough),
        symptom(Patient, sneezing),
        symptom(Patient, runny_nose).

hypothesis(Patient, flu) :-
        symptom(Patient, fever),
        symptom(Patient, headache),
        symptom(Patient, body_ache),
        symptom(Patient, conjunctivitis),
        symptom(Patient, chills),
        symptom(Patient, sore_throat),
        symptom(Patient, runny_nose),
        symptom(Patient, cough).

hypothesis(Patient, mumps) :-
        symptom(Patient, fever),
        symptom(Patient, swollen_glands).

hypothesis(Patient, chicken_pox) :-
        symptom(Patient, fever),
        symptom(Patient, chills),
        symptom(Patient, body_ache),
        symptom(Patient, rash).
```

```
write_list([]).
write_list([Term|Terms]) :-
        write(Term),
        write_list(Terms).
```



```
mmcoe@mmcoe-System-Product-Name:~$ cd Desktop
mmcoe@mmcoe-System-Product-Name:~/Desktop$ swipl -o asn6 -c asn6.pl
% Disabled autoloading (loaded 38 files)
% Disabled autoloading (loaded 0 files)
mmcoe@mmcoe-System-Product-Name:~/Desktop$ ./asn6
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- go.
What is the patient's name? pranav.
pranav, do you have a fever (y/n) ?|: y.
pranav, do you have a headache (y/n) ?|: n.
pranav, do you have a cough (y/n) ?|: n.
pranav, do you have swollen_glands (y/n) ?|: n.
pranav, do you have chills (y/n) ?|: y.
pranav, do you have a body_ache (y/n) ?|: y.
pranav, do you have a rash (y/n) ?|: y.
pranav, probably has chicken_pox.
true .
```