1)

a) Let A = I am Here

Let B = Jane has a cat Named Patty

Let C = John has a dog named Curiosity

Let D = Curiosity Killed the Cat

Given

$\neg A \vee B$

$\neg B \vee C$

$\neg C \vee D$

A

$$A \wedge (A \rightarrow B) \wedge (B \rightarrow C) \wedge (C \rightarrow D)$$

A	B	C	D	$\neg A \vee B$	$\neg B \vee C$	$\neg C \vee D$
T	T	T	T	T	T	T
T	T	T	F	T	T	F
T	T	F	T	T	F	T
T	T	F	F	T	F	T
T	F	T	T	F	T	T
T	F	T	F	F	T	F
T	F	F	T	F	T	T
T	F	F	F	F	T	T
F	T	T	T	T	T	T
F	T	T	F	T	T	F
F	T	F	T	T	F	T
F	T	F	F	T	F	T
F	F	T	T	T	T	T
F	F	T	F	T	T	F
F	F	F	T	T	T	T
F	F	F	F	T	T	T

b) $A \wedge (\neg A \vee B) \wedge (\neg B \vee C) \wedge (\neg C \vee D)$

$A \wedge (A \rightarrow B) \wedge (B \rightarrow C) \wedge (C \rightarrow D)$

$A \wedge (A \rightarrow C) \wedge (C \rightarrow D)$

$A \wedge (A \rightarrow D) \Leftrightarrow D$

Curiosity Killed the Cat logically follows

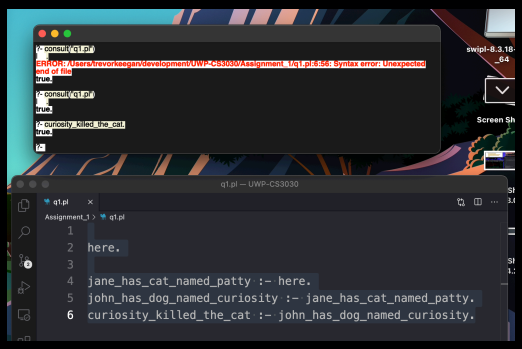
c)

$(\neg A \vee B) = A \rightarrow B$ B :- A

$(\neg B \vee C) = B \rightarrow C$ C :- B

$(\neg C \vee D)$ $C \rightarrow D$ D :- C

A




```
here.

jane_has_cat_named_patty :- here.
john_has_dog_named_curiosity :- jane_has_cat_named_patty.
curiosity_killed_the_cat :- john_has_dog_named_curiosity.
```

2)

Assignment_1 > q2.pl

```
%Rules into predicate logic - A
%   clear(Block_A) is true if no block is placed on top of Block_A
%   on(Block_A, Block_B) is true if Block_A is placed on top of Block_B
%
%   move Block_A from Block_B to Block_C
%   clear(Block_A), clear(Block_C), Block_C \= table, on(Block_A, Block_B) -> on(Block_A, Block_C)
%
%   move Block_A from Block_B to table
%   clear(Block_A), on(Block_A, Block_B) -> on(Block_A, table)
%


% B
%   on(A,C) :- clear(A), clear(C), C \= table, on(A, B)
%   on(A,table) :- clear(A), on(A,B)

% D

% initial state
on(a,b).
on(b,table).
on(c,table).
clear(a).
clear(c).


% discovered format function at https://stackoverflow.com/questions/34635689/output-formatting-in-prolog
% Block X is on Block Y provided that X is clear, Y is clear, Y is on some block Z, and Y is not the table
on(X, Y) :- clear(X), clear(Y), on(X, Z), neq_table(Y), format("move ~w from ~w to ~w ~n", [X, Y, Z]).

% Block C is clear provided that A is clear, A is on some block X, and C is not the table
clear(C) :- clear(A), on(A, Z), neq_table(C),format("move ~w from ~w to the table ~n", [A, Z]) .

% Block C is clear provided that A is clear, A is on some block X, and C is not the table
on(A,table) :- clear(A), on(A, Z), neq_table(C), format("move ~w from ~w to the table ~n", [A, Z]) .
neq_table(X) :- X \= table.

```

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.3.18)
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).

?- working_directory(_,"/Users/trevorkeegan/development/UWP-CS3030/Assignment_1").
true.

?- consult("q2.pl").
Warning: /Users/trevorkeegan/development/UWP-CS3030/Assignment_1/q2.pl:28:
Warning:    Clauses of on/2 are not together in the source-file
Warning:    Earlier definition at /Users/trevorkeegan/development/UWP-CS3030/Assignment_1/q2.pl:19
Warning:    Current predicate: clear/1
Warning:    Use :- discontiguous on/2. to suppress this message
Warning: /Users/trevorkeegan/development/UWP-CS3030/Assignment_1/q2.pl:31:
Warning:    Clauses of clear/1 are not together in the source-file
Warning:    Earlier definition at /Users/trevorkeegan/development/UWP-CS3030/Assignment_1/q2.pl:22
Warning:    Current predicate: on/2
Warning:    Use :- discontiguous clear/1. to suppress this message
Warning: /Users/trevorkeegan/development/UWP-CS3030/Assignment_1/q2.pl:34:
Warning:    Singleton variables: [C]
Warning: /Users/trevorkeegan/development/UWP-CS3030/Assignment_1/q2.pl:34:
Warning:    Clauses of on/2 are not together in the source-file
Warning:    Earlier definition at /Users/trevorkeegan/development/UWP-CS3030/Assignment_1/q2.pl:19
Warning:    Current predicate: clear/1
Warning:    Use :- discontiguous on/2. to suppress this message
true.

?- on(c,b).
move a from b to the table
move c from b to table
true .
```

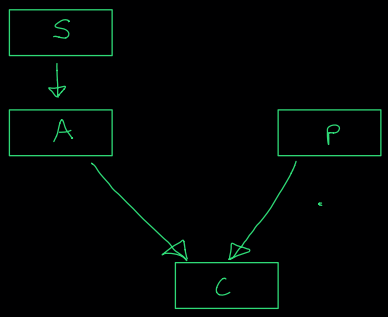
3)

Let S = Person is Symptomatic

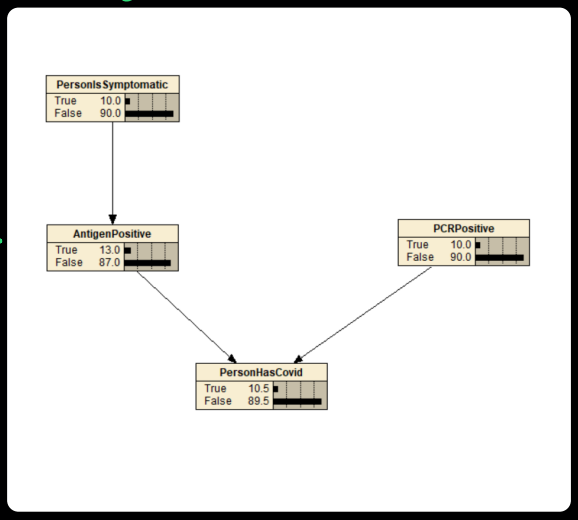
Let A = Person Has Positive Antigen Test

Let P = Person Has Positive PCR Test

Let C = Person Got Covid



Compiled



Symptoms → 11.7% chance of Having Covid

Force Symptomatic @ true