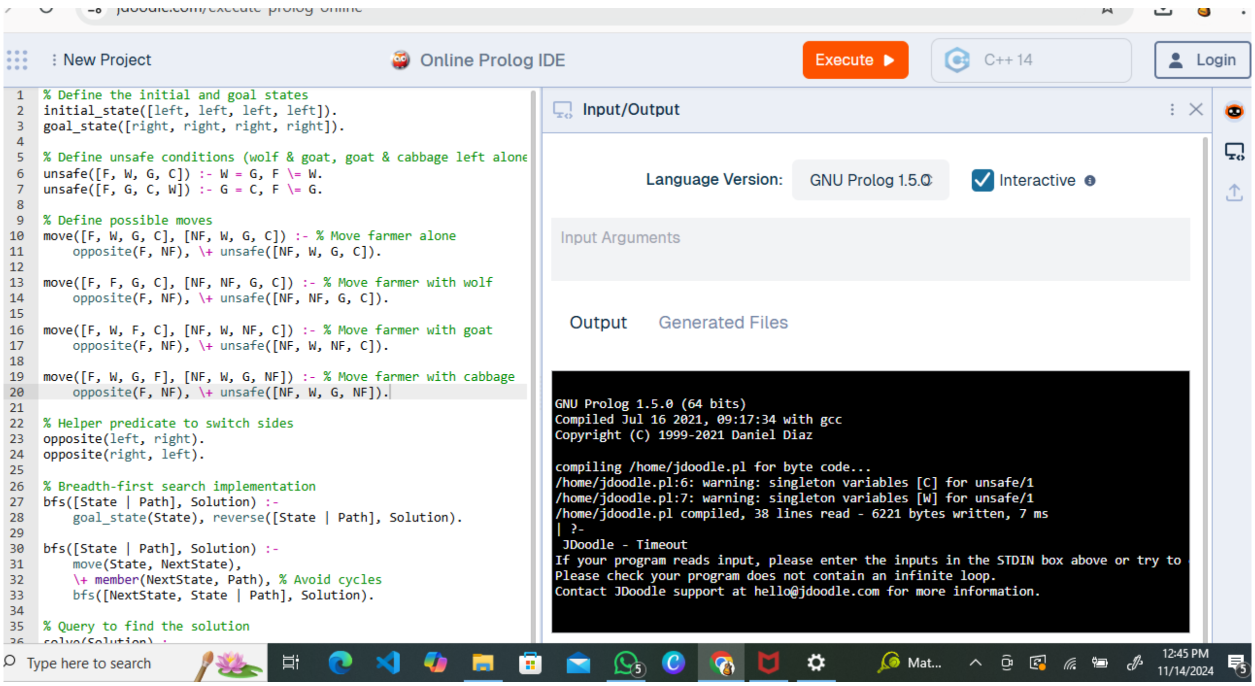
KAHENI PETER

P101/1740G/21

ASSIGMENT

LOGIC PROGRAMMING



```prolog
% Define the initial and goal states
initial_state([left, left, left, left]).
goal_state([right, right, right, right]).

% Define unsafe conditions (wolf & goat, goat & cabbage left alone
unsafe([F, W, G, C]) :- W = G, F \= W.
unsafe([F, G, C, W]) :- G = C, F \= G.

% Define possible moves
move([F, W, G, C], [NF, W, G, C]) :- % Move farmer alone
    opposite(F, NF), \+ unsafe([NF, W, G, C]).

move([F, F, G, C], [NF, NF, G, C]) :- % Move farmer with wolf
    opposite(F, NF), \+ unsafe([NF, NF, G, C]).

move([F, W, F, C], [NF, W, NF, C]) :- % Move farmer with goat
    opposite(F, NF), \+ unsafe([NF, W, NF, C]).

move([F, W, G, F], [NF, W, G, NF]) :- % Move farmer with cabbage
    opposite(F, NF), \+ unsafe([NF, W, G, NF]).

% Helper predicate to switch sides
opposite(left, right).
opposite(right, left).

% Breadth-first search implementation
bfs([State | Path], Solution) :-
    goal_state(State), reverse([State | Path], Solution).

bfs([State | Path], Solution) :-
    move(State, NextState),
    \+ member(NextState, Path), % Avoid cycles
    bfs([NextState, State | Path], Solution).

% Query to find the solution
solve(Solution) :-
```

Language Version: GNU Prolog 1.5.0   ☑ Interactive

```
GNU Prolog 1.5.0 (64 bits)
Compiled Jul 16 2021, 09:17:34 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling /home/jdoodle.pl for byte code...
/home/jdoodle.pl:6: warning: singleton variables [C] for unsafe/1
/home/jdoodle.pl:7: warning: singleton variables [W] for unsafe/1
/home/jdoodle.pl compiled, 38 lines read - 6221 bytes written, 7 ms
| ?-
 JDoodle - Timeout
If your program reads input, please enter the inputs in the STDIN box above or try to
Please check your program does not contain an infinite loop.
Contact JDoodle support at hello@jdoodle.com for more information.
```