**CSCE 5450 PROGRAMMING LANGUAGES**

**PROJECT-3**

% Graph representation: edge(Node1, Node2)

edge(a, b).

edge(a, c).

edge(b, d).

edge(c, e).

edge(c, f).

edge(e, f).

% Breadth-First Search

bfs(Start, Goal, Path) :-

bfs\_queue([[Start]], Goal, [], Path).

% Helper predicate for BFS using a queue

bfs\_queue([[Goal|Rest] | \_], Goal, \_, Path) :-

reverse([Goal|Rest], Path),

write\_path(Path).

bfs\_queue([[Current|Rest] | OtherPaths], Goal, Visited, Path) :-

findall([Next, Current|Rest],

(edge(Current, Next), \+ member(Next, [Current|Visited])),

NewPaths),

append(OtherPaths, NewPaths, Queue),

bfs\_queue(Queue, Goal, [Current|Visited], Path).

% Print the path

write\_path([]).

write\_path([Node]) :-

write(Node).

write\_path([Node|Rest]) :-

write(Node),

write(' -> '),

write\_path(Rest).

% Test cases

test\_case1 :-

bfs(a, f, Path),

writeln('Path from a to f:'),

writeln(Path). % Expected output: [a, c, f]

test\_case2 :-

bfs(b, f, Path),

writeln('Path from b to f:'),

writeln(Path). % Expected output: [b, d, c, f]

:- initialization(test\_case1).

:- initialization(test\_case2).

