EXPERIMENT:27 Write a Prolog Program to implement Best First Search algorithm

PROGRAM:

```
% edge(Node1, Node2)

edge(a, b). edge(a, c).

edge(b, d). edge(b, e).

edge(c, e). edge(d, goal). edge(e, goal).


% heuristic(Node, H)

h(a, 7). h(b, 6). h(c, 2). h(d, 1). h(e, 0). h(goal, 0).


% Best-First Search

best_first(Start, Goal, Path) :- bfs([[Start]], Goal, Path).


bfs([[Goal|Rest]|_], Goal, Path) :-
    reverse([Goal|Rest], Path).


bfs([CurrentPath|OtherPaths], Goal, FinalPath) :-
    CurrentPath = [Node|_],
    findall([Next,Node|CurrentPath],
            (edge(Node, Next), \+ member(Next, CurrentPath)),
```

```
        NewPaths),
    append(OtherPaths, NewPaths, Paths),
    sort_paths(Paths, SortedPaths),
    bfs(SortedPaths, Goal, FinalPath).

% sort paths by heuristic of first node in each path
sort_paths(Paths, Sorted) :-
    map_list_to_pairs(path_heuristic, Paths, Pairs),
    keysort(Pairs, SortedPairs),
    pairs_values(SortedPairs, Sorted).

path_heuristic([Node|_], H) :- h(Node, H).
```

OUTPUT: