18BECE30558
TITHI PATEL

PRACTICAL-10

Program: Write a program to implement A* Algorithm.

```
%%%
%%%
%%% Nodes have form S#D#F#A
%%% where S describes the state or
configuration
%%% D is the depth of the node
%%% F is the evaluation function value
%%% A is the ancestor list for the node
: op(400,yfx,'#'). /* Node builder notation */
solve(State,Soln) : f_function(State,0,F),

search([State#0#F#[]],S), reverse(S,Soln).
f_function(State,D,F) : h_function(State,H),

F is D + H.

search([State#_#_#Soln|_], Soln) : goal(State).
search([B|R],S) : expand(B,Children),
insert_all(Children,R,Open),
search(Open,S).
insert_all([F|R],Open1,Open3) : insert
(F,Open1,Open2),

insert_all(R,Open2,Open3).

insert_all([],Open,Open).
insert(B,Open,Open) : repeat_node(B,Open), ! .
insert(B,[C|R],[B,C|R]) : cheaper(B,C), ! .
insert(B,[B1|R],[B1|S]) : insert(B,R,S), !.
insert(B,[],[B]).
repeat_node(P#_#_#_, [P#_#_#_|_]).
cheaper( _#_#F1#_ , _#_#F2#_ ) : F1 < F2.
expand(State#D#_#S,All_My_Children) :
bagof(Child#D1#F#[Move|S],
(D1 is D+1,
```

move(State,Child,Move),f_function(Child,D1,F)), All_My_Children