**VISHWAKARMA INSTITUTE OF TECHNOLOGY**

COMPUTER ENGINEERING

**Name: Abhinav Mahajan**

**Division: TY-C**

**Roll No: 15**

**Subject: Artificial Intelligence (AI)**

**LAB ASSIGNMENT NO – 3**

Implementation of **Informed Strategies** for **8-Puzzle Game**.

**A\* Algorithm:**

**Approach:**

**Code:**

*#include* <bits/stdc++.h>

using namespace std;

void findZero(vector<vector<int>> board, int &x, int &y){

*for* (int i = 0; i < board.size(); i++){

*for* (int j = 0; j < board.size(); j++){

*if* (board[i][j] == 0){

                x = i;

                y = j;

*return*;

            }

        }

    }

}

void printBoard(vector<vector<int>> board){

*for* (int i = 0; i < board.size(); i++){

*for* (int j = 0; j < board.size(); j++){

            cout << board[i][j] << " ";

        }

        cout << endl;

    }

    cout << endl;

}

bool isGoalState(vector<vector<int>> &board, vector<vector<int>> &goal) {

*return* board == goal;

}

int findMisplacedTiles(vector<vector<int>> &board, vector<vector<int>> &goal){

    int count = 0;

*for* (int i = 0; i < board.size(); i++){

*for* (int j = 0; j < board.size(); j++){

*if* (board[i][j] != goal[i][j])

                count++;

        }

    }

*return* count;

}

void aStar(vector<vector<int>> &board, vector<vector<int>> &goal, int depth, int x, int y){

    priority\_queue<pair<int, vector<vector<int>>>, vector<pair<int, vector<vector<int>>>>, greater<pair<int, vector<vector<int>>>>> pq;

    int g = depth;

    int h = findMisplacedTiles(board, goal);

    pq.push({(g + h), board});

*while* (!pq.empty()){

        vector<vector<int>> curr = pq.top().second;

        pq.pop();

        printBoard(curr);

        int x, y;

        findZero(curr, x, y);

*if* (isGoalState(curr, goal)){

            cout << "Goal State Reached" << endl;

*return*;

        }

        int dx[] = {0, 0, -1, 1};

        int dy[] = {1, -1, 0, 0};

*for* (int i = 0; i < 4; i++){

            int newX = x + dx[i];

            int newY = y + dy[i];

*if* (newX >= 0 && newX < curr.size() && newY >= 0 && newY < curr.size()){

                swap(curr[x][y], curr[newX][newY]);

                g = depth + 1;

                h = findMisplacedTiles(curr, goal);

                pq.push({(g + h), curr});

                swap(curr[x][y], curr[newX][newY]);

            }

        }

    }

*return*;

}

int main() {

    vector<vector<int>> initial = {

        {2, 8, 3},

        {1, 6, 4},

        {7, 0, 5}

    };

    vector<vector<int>> goal = {

        {1, 2, 3},

        {8, 0, 4},

        {7, 6, 5}

    };

    int x, y;

    findZero(initial, x, y);

*// vector<vector<vector<int>>> ans;*

    aStar(initial, goal, 0, x, y);

*return* 0;

}

**Output:**

