***ARTIFICIAL INTELLIGENCE PROJECT***

***8 PUZZLE PROBLEM***

SUBMITTED BY:

NAME: EKANSH SRIVASTAVA

REGISTRATION NO.:11801743

SECTION: K18KK / GROUP-1

ROLL NO : 30

In partial fulfilment for the requirements of award of the degree “B.Tech CSE(Hons.)”

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

**LOVELY PROFESSIONAL UNIVERSITY**

**PHAGWARA, PUNJAB**

**INTRODUCTION**

The 8-puzzle problem is a puzzle invented and popularized by Noyes Palmer Chapman in the 1870s. It is played on a 3-by-3 grid with 8 square blocks labeled 1 through 8 and a blank square. Your goal is to rearrange the blocks so that they are in order. You are permitted to slide blocks horizontally or vertically into the blank square. The following shows a sequence of legal moves from an initial board position to the goal position

**Objective**

Solving the 8 puzzle using A\*(a star)

**AIM**

Through this project, we would be able to analyze and implement a popular algorithm in a gaming scope. This would ensure the understanding of similar algorithms in different games and other fields.

**ABSTRACT**

N-puzzle problem has been one of the basic problem since the beginning of artificial intelligence. The most popular version of n-puzzle among people is 8- puzzle problem. It consists of an area divided into 3x3 grid containing 8 numbered (to identify) tiles and one empty grid. We are given an initial state and we have to reach the goal state which is also specified. In this project, we have used informed search method called a\*algorithm to solve the puzzle. Various heuristic involved in the informed search like number of misplaced tiles, Manhattan distance were analyzed; Manhattan distance being one of the most popular ones.

**Implementation**

Using command line arguments the program takes two arguments that describes the program which heuristic function to take into account and what format of output is required by the user

2 heuristic functions have been implemented in the program:

* Hamming priority function.

The number of blocks in the wrong position, plus the number of moves made so far to get to the state. A state with a small number of blocks in the wrong position is close to the goal state, and we prefer a state that have been reached using a small number of moves.

* Manhattan priority function.

The sum of the distances (sum of the vertical and horizontal distance) from the blocks to their goal positions, plus the number of moves made so far to get to the state.

2 output formats have been implemented:

* Each state of the puzzle being printed.
* Each move to get to the final state being printed

The main function check all the CLI arguments for conditions and then takes 2 lines as input with each tile separated with a ‘ ‘ (space).

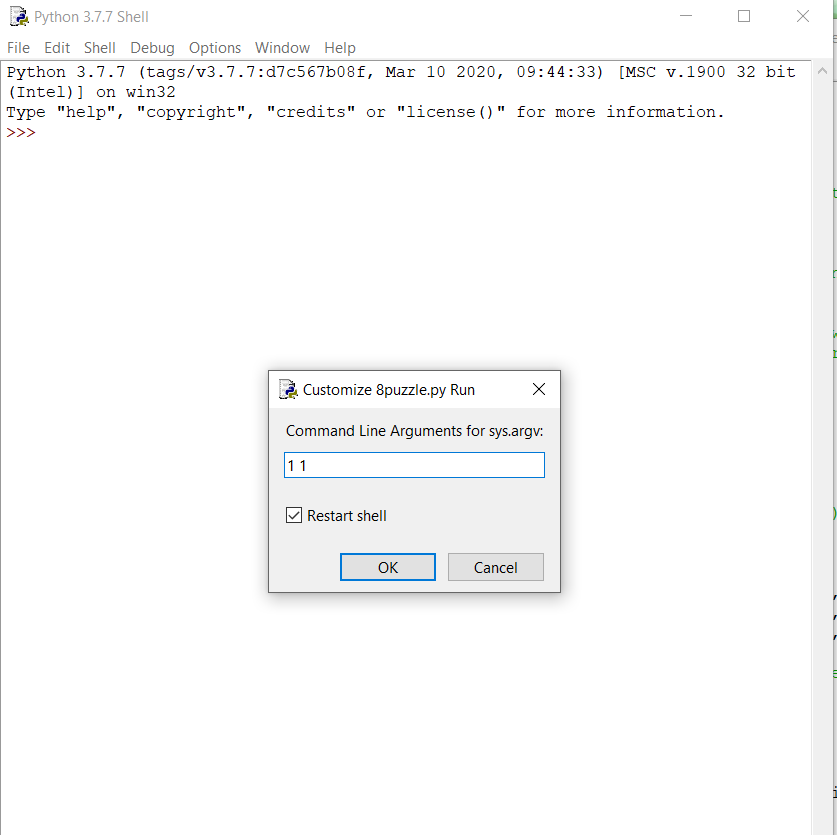
1. The initial position of the puzzle to be solved
2. The final position of the puzzle

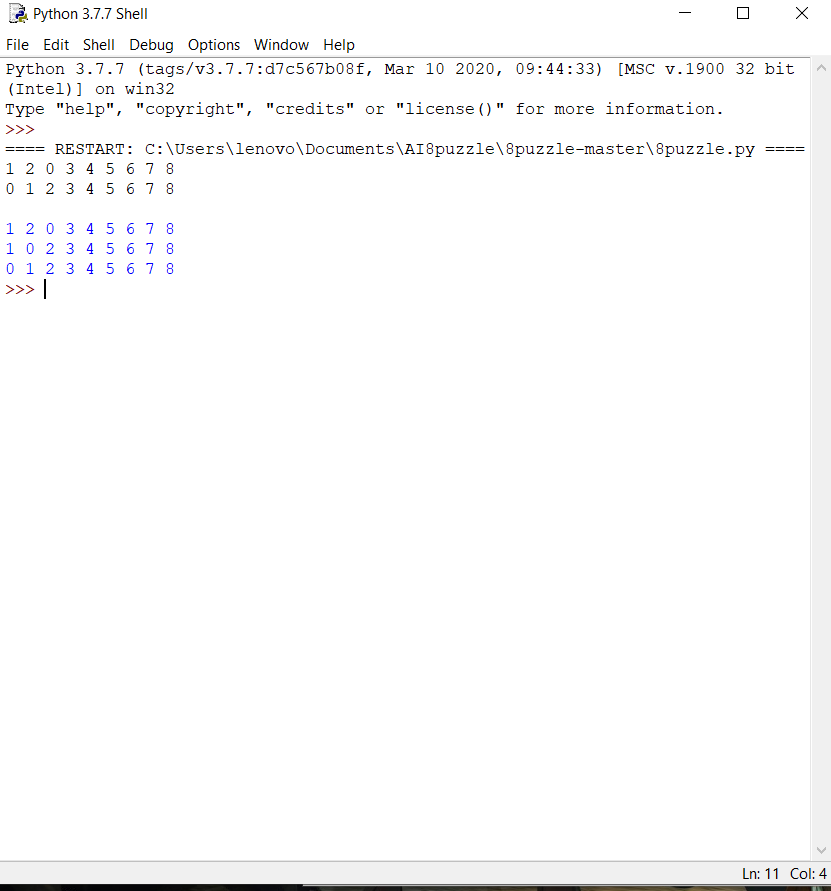
Another function sets it into grid format and the output is printed via calling the Astar function that takes in initial state, heuristics to be followed and the goal state and prints the output according to the specified format

**Code Snippet**

The code can be checked in the github link at the end of file

**OUTPUT**





**GITHUB LINK**

<https://github.com/ekansh9696/8-puzzle-using-A->