Consider you have an agent for an **Eight puzzle problem**. The goal of an agent is to complete the Eight puzzle in **fixed number of moves**. Your implementation should be modular so that the sensors, actuators and environment characteristics (size etc.) can be easily changed. You need to implement a graph based search agent with visited state check. You can use the provided agents.py file which contains the abstract implementation of agent and environment. You are required to formulate this problem by defining the four parameters:

1. Initial State
2. Successor Function
3. Goal Test
4. Path cost

**Problem Description:**

- • Available actions are Left, Right, Up, Down, NoOp.
    - o Left, Right, Up, Down and NoOp have the same meaning as discussed in the class

- The termination criterion is that the agent runs out of moves or the puzzle is completed.
- Environment announces when this criterion is met and stops the execution.
- The performance of an agent is calculated after the termination criteria is met. The performance measure of an agent is the (# of correctly placed items) / (number of steps used). Note that there will be early termination if puzzle is solved before expiry of maximum number of moves.
- The environment is deterministic and partially observable.
- Agent knows the size of puzzle (grid n x n) and the content of the cell they land in, location of the landing cell (coordinates) is not known.
- The perception is given by the environment and includes, cell coordinates and if the current piece in the cell is rightly placed or not.
- Starting position of the agent is random and not known beforehand plus puzzle contents are randomized at each start.

**Submission**

Assignment must be submitted in the google classroom.
Submission other than google classroom won't not be accepted.
You are required to submit a python (version 3 compatible) file that is called in the following manner.

- First argument is puzzle size and second argument is the maximum number of moves allowed (70).
- Name of the file uses pattern "Lab4_<ROLLNO>.py"
- python3 Lab4_17I-0000.py

**output should be like :**

No of correct pieces = 30, no of moves utilized = 150