Experiment No. 1

Name : Aniket Narbariya

UID : 2019130043

Class : TE COMPS Batch C

Subject : AIML

Aim : To create an Intelligent Agent using AI.

Theory :

An intelligent agent is a program that can make decisions or perform a service based on its environment, user input and experiences. These programs can be used to autonomously gather information on a regular, programmed schedule or when prompted by the user in real time.

The PEAS system delivers the performance measure with respect to the environment, actuators and sensors of the respective agent. Most of the highest performing agents are Rational Agents.

Rational Agent: The rational agent considers all possibilities and chooses to perform the highly efficient action. For example it chooses the shortest path with low cost for high efficiency.

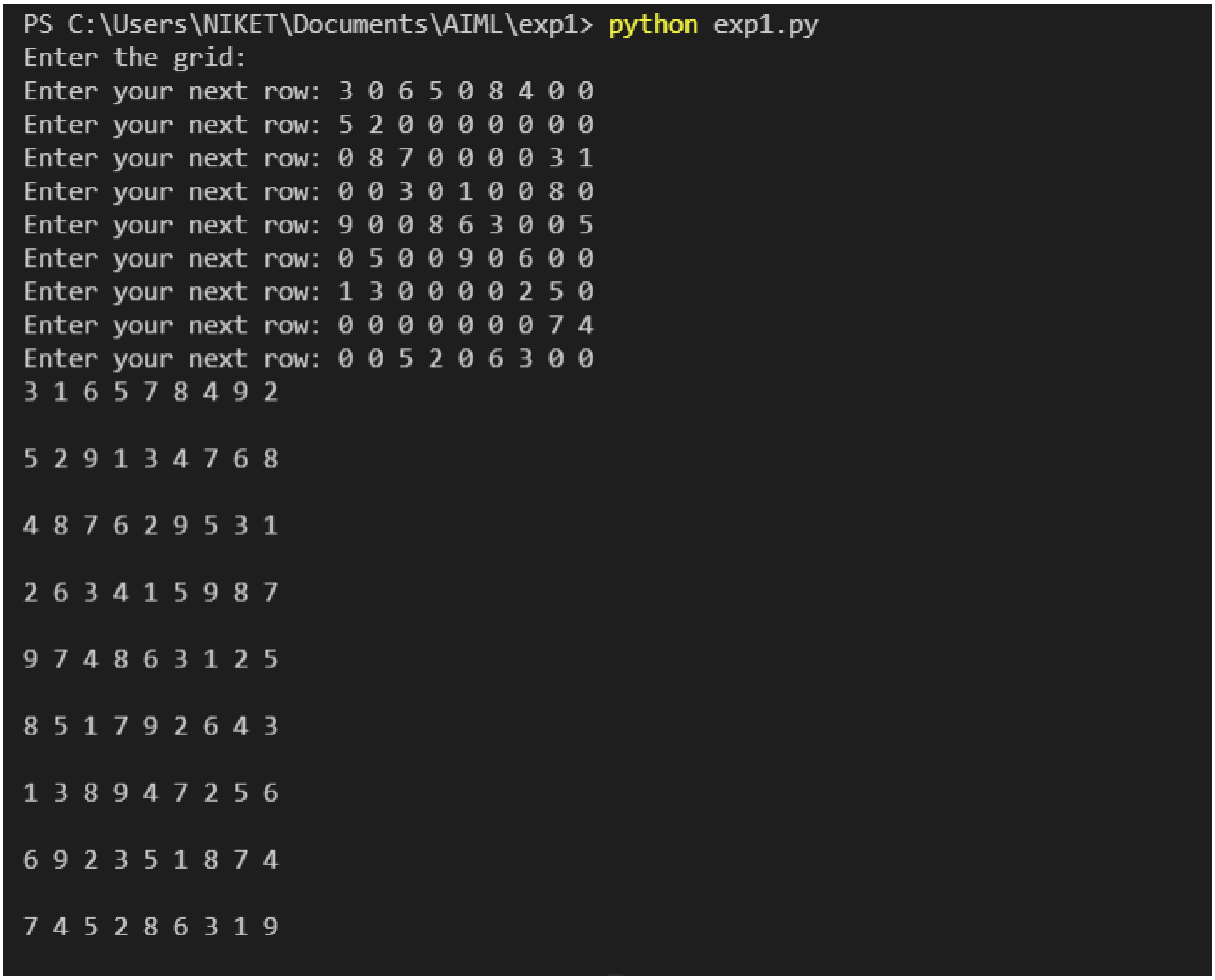
PEAS stands for *Performance measure, Environment, Actuator, Sensor*.

1. **Performance Measure:** Performance measure is the unit to define the success of an agent. Performance varies with agents based on their different precept.
2. **Environment**: Environment is the surrounding of an agent at every instant. It keeps changing with time if the agent is set in motion. There are 5 major types of environments:
   - Fully Observable & Partially Observable
   - Episodic & Sequential
   - Static & Dynamic
   - Discrete & Continuous
   - Deterministic & Stochastic
3. **Actuator**: Actuator is a part of the agent that delivers the output of an action to the environment.
4. **Sensor**: Sensors are the receptive parts of an agent which takes in the input for the agent.

Program :

```
def print_grid(grid):
    for i in range(9):
        for j in range(9):
            print(grid[i][j],end=" ")
        print ('\n')

def find_unassigned_location(grid, l):
    for row in range(9):
        for col in range(9):
            if(grid[row][col]== 0):
                l[0]= row
                l[1]= col
                return True
    return False

def used_in_row(grid, row, num):
    for i in range(9):
        if(grid[row][i] == num):
            return True
    return False

def used_in_col(grid, col, num):
    for i in range(9):
        if(grid[i][col] == num):
            return True
    return False

def used_in_box(grid, row, col, num):
    start_row = row - row%3
    start_col = col - col%3
    for i in range(3):
        for j in range(3):
            if(grid[i + start_row][j + start_col] == num):
                return True
    return False

def is_safe(grid, row, col, num):
    return not used_in_row(grid, row, num) and not used_in_col(grid, col, num) and not used_in_box(grid, row,col, num)

def solve_sudoku(grid):

     #Here, l is a list which keeps our next unassigned location 
    l =[0, 0]
    
    # If there is no unassigned location,then we return True 
    if(not find_unassigned_location(grid, l)):
        return True
    
    #Get the row number and column number of next unassigned location
    row = l[0]
    col = l[1]
    
    # consider digits 1 to 9
    for num in range(1, 10):
        if(is_safe(grid,row, col, num)):
            grid[row][col]= num
            if(solve_sudoku(grid)):
                return True
            grid[row][col] = 0   
    return False
#Define our grid
grid = []

print("Enter the grid:")
for i in range(1,10):
    row = list(map(int, input("Enter your next row: ").strip().split()))
    grid.append(row)
if(solve_sudoku(grid)):
    print_grid(grid)
else:
    print("No solution")
```

Output:



```
PS C:\Users\NIKET\Documents\AIML\exp1> python exp1.py
Enter the grid:
Enter your next row: 3 0 6 5 0 8 4 0 0
Enter your next row: 5 2 0 0 0 0 0 0 0
Enter your next row: 0 8 7 0 0 0 0 3 1
Enter your next row: 0 0 3 0 1 0 0 8 0
Enter your next row: 9 0 0 8 6 3 0 0 5
Enter your next row: 0 5 0 0 9 0 6 0 0
Enter your next row: 1 3 0 0 0 0 2 5 0
Enter your next row: 0 0 0 0 0 0 0 7 4
Enter your next row: 0 0 5 2 0 6 3 0 0
3 1 6 5 7 8 4 9 2

5 2 9 1 3 4 7 6 8

4 8 7 6 2 9 5 3 1

2 6 3 4 1 5 9 8 7

9 7 4 8 6 3 1 2 5

8 5 1 7 9 2 6 4 3

1 3 8 9 4 7 2 5 6

6 9 2 3 5 1 8 7 4

7 4 5 2 8 6 3 1 9
```

Conclusion :

I learned about Intelligent agents . I was able to create sudoku solver as an Intelligent Agent. PEAS of AI – Sudoku Solver are

P – Solve the puzzle

E – Sudoku board

A – Actions to be taken according to the bounds

S – No Sensors