Documentation

1. Theoretical Analysis:

The goal of this program is to simulate an agent that learns the probability distribution of object placements on a 6x6 grid, based on observations from multiple sample grids. The agent then traverses a test grid by moving to the highest unvisited probability position, counting successful object findings and errors (locations without objects).

The simulation employs random grid generation to build a probability distribution, which the agent uses to guide its traversal. The agent's traversal is animated, highlighting successful and error placements as it moves.

2. Data Structures:

- Grids (List of 2D numpy arrays): Stores multiple 6x6 grids, each representing a possible object placement pattern.
These grids are generated randomly.
- Probability Grid (2D numpy array): Stores the probability of an object being in each cell based on training grids.
Calculated as the sum of all grids, normalized by the number of grids.

- Test Grid (2D numpy array): Represents a single 6x6 grid on which the agent is tested. Randomly generated similarly to the training grids.
- Agent Position (Tuple): Tracks the agent's current position within the grid during traversal.
- Visited (2D numpy array of bools): Marks cells as visited to avoid revisiting.
- Path (List of Tuples): Records the agent's path as it moves through the grid.
- Success Count (int): Tracks successful finds in the test grid.
- Error Count (int): Tracks error moves (where no object is found).

3. Algorithm to Function/Method Representation:

1. Grid Generation and Probability Calculation:
   - Generate multiple grids with random placements.
   - For each cell, calculate the probability of containing an object based on the number of training grids.

2. Agent Initialization and Traversal:
   - Initialize the agent at the cell with the highest probability in the probability grid.
   - Traverse unvisited cells in descending order of their probabilities.
   - Count successes (finding an object) and errors (no object) for each cell visited.

3. Visualization (Using matplotlib):

- Display the probability grid with annotated values.

- Use animation to update the agent's current cell, marking successes and errors visually.

- At the end, display success and error percentages.

4.Implementation:

The program is structured as follows:

- Grid Generation: Generates 50 grids of random object placements.

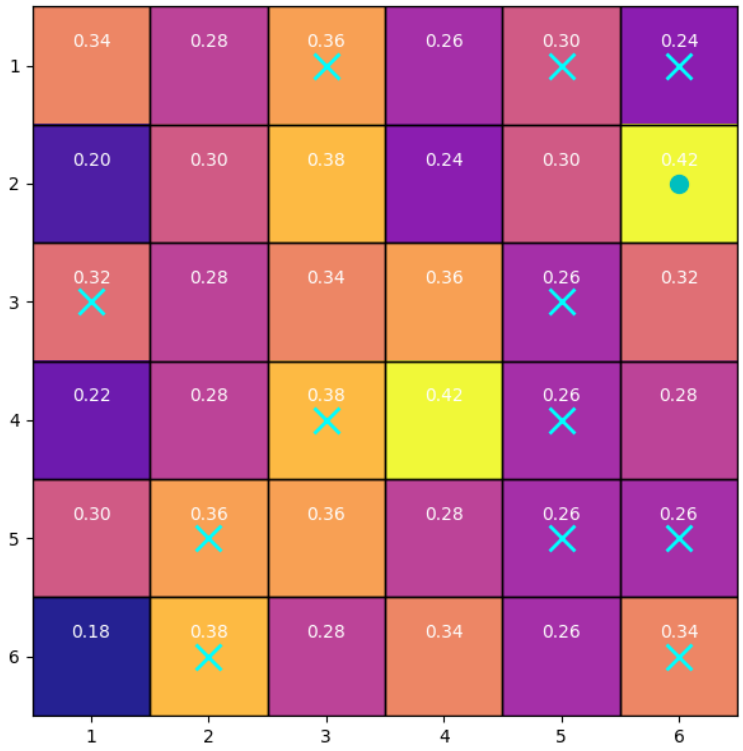
- Probability Calculation: Iterates over each generated grid to count object placements, converting the counts into probabilities by dividing by the total number of grids.

- Agent Initialization and Traversal: The agent starts at the cell with the highest probability in the probability grid. The agent then moves to unvisited cells with the highest remaining probabilities, recording successes and errors.

- Animation and Visualization: An animated grid displays the agent's path with markers for successes (black squares) and errors (magenta circles), the agent's current position (cyan circle), and annotated probability values. Success/error counts update during the traversal.

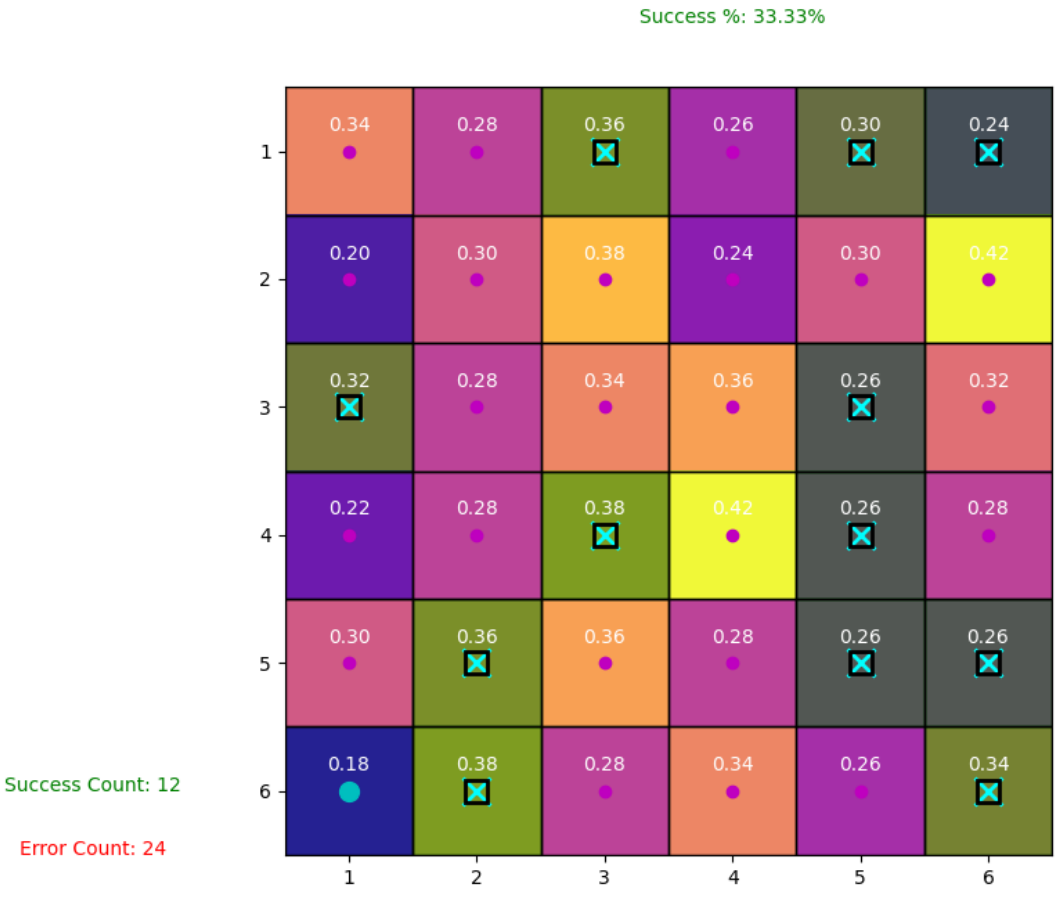
5.Input Test Cases Format:

(A)Before traversing:

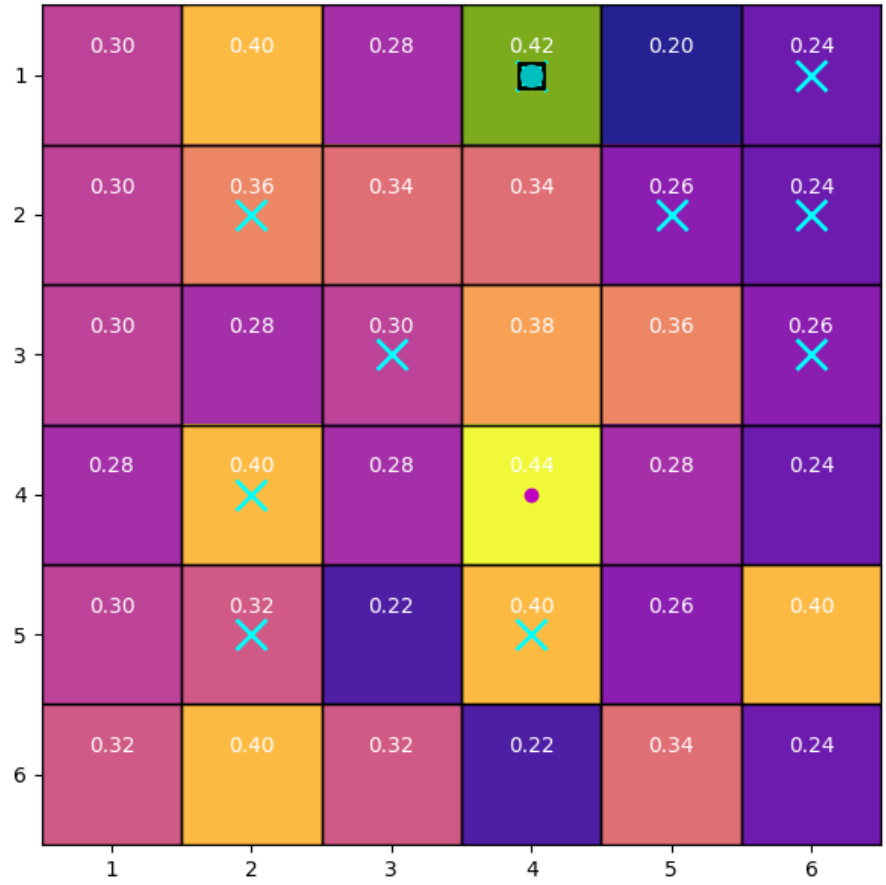


6.Output Format:

(A)After traversing:



(B)Before traversing:



(B)After traversing: