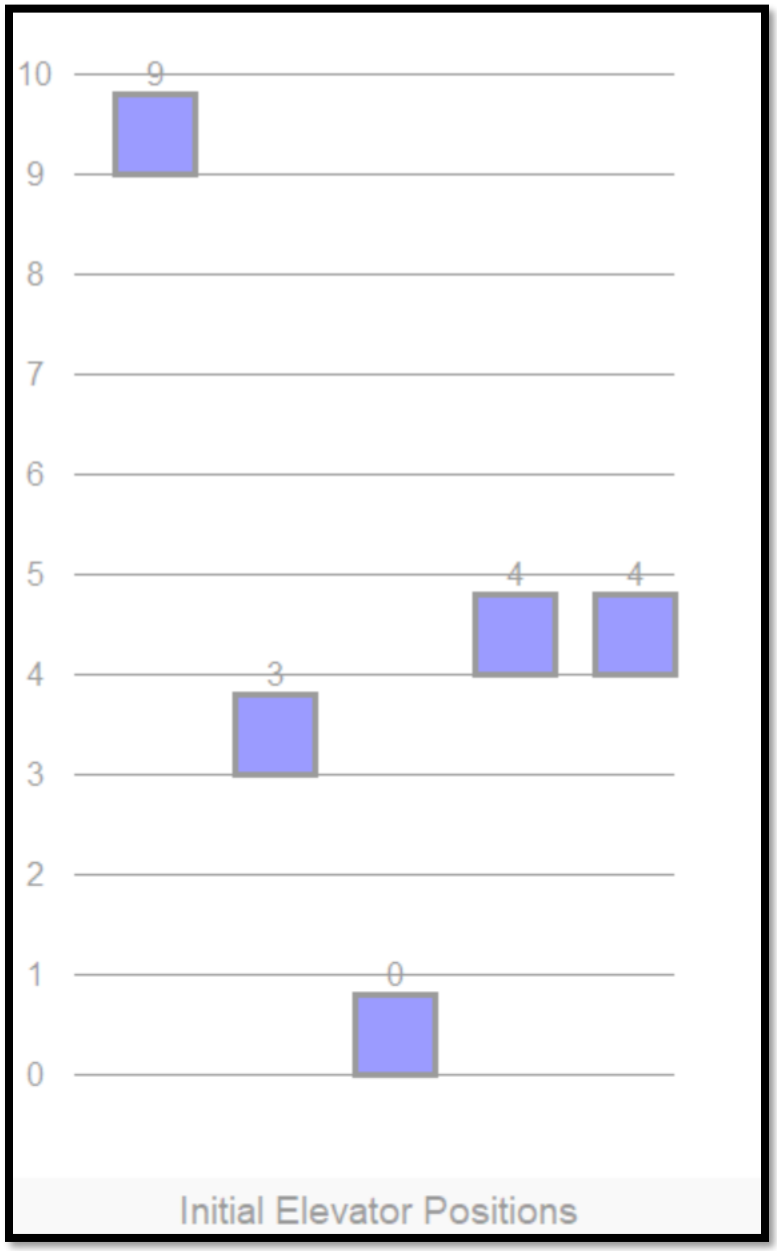
Implementation Steps

1. Initialize Simulation
2. Create the ElevatorSimulation class to manage floors, elevators, and positions.
3. Randomly assign initial elevator positions and store them for visualization.
4. Define Elevator Movement
5. Implement move_elevator to move elevators step by step to a target floor, recording each step.
6. Handle Elevator Requests
7. Use handle_calls to process floor requests by selecting the nearest elevator and moving it.
8. Build the GUI
9. Create the ElevatorApp class with Tkinter to display controls and visualize positions.
10. Design Visualization
11. Use a canvas to draw floors and elevator positions dynamically.
12. Add Navigation
13. Include "Next" and "Previous" buttons for navigating through recorded positions, updating the canvas.
14. Connect Simulation to GUI
15. Process user inputs, handle elevator requests, and enable navigation buttons once data is available.
16. Run Application
17. Initialize the simulation and GUI, then start the Tkinter event loop.

Implementation Output

```
"A:\MUJ\SEMESTER 5\AI_SOFT_COMPUTING_LAB\labx\pythonProject\.venv\Scripts\pytho
Enter the number of floors in the building (0 to n): 10
Press Enter to continue after observing the initial elevator positions...
```



```
How many floors do you want to call the lift for? 6
Enter the floor number for call 1: 0
Enter the floor number for call 2: 4
Enter the floor number for call 3: 8
Enter the floor number for call 4: 2
Enter the floor number for call 5: 10
Enter the floor number for call 6: 3
```