

```
#!/usr/bin/env python3
```

```
"""
```

Virtual Parking Assistant (ASCII Grid) – Final Working Implementation

This script provides a command-line ASCII grid simulation of a parking lot.

It supports spot allocation, release, and real-time visualization with collision avoidance.

```
"""
```

Class ParkingLot:

```
    Def __init__(self, rows=5, cols=10):
```

```
        Self.rows = rows
```

```
        Self.cols = cols
```

```
        # Initialize the grid with '.' Denoting vacant spots
```

```
Self.grid = [['.' For _ in range(cols)] for _ in range(rows)]
```

```
Def display(self):
```

```
    """Display current parking lot state."""
```

```
    Print("\nParking Lot Status:")
```

```
    Print(" " + " ".join(str(i) for i in range(self.cols)))
```

```
    For idx, row in enumerate(self.grid):
```

```
        Print(f"{idx} " + " ".join(row))
```

```
    Print("Legend: '.' = vacant, 'X' = occupied\n")
```

```
Def find_nearest_spot(self):
```

```
    """Find the nearest available parking spot (top-left priority)."""
```

```
    For r in range(self.rows):
```

```
        For c in range(self.cols):
```

```
    If self.grid[r][c] == '.':
```

```
        Return (r, c)
```

```
    Return None # No available spots
```

```
Def park_vehicle(self):
```

```
    """Allocate the nearest available spot."""
```

```
    Spot = self.find_nearest_spot()
```

```
    If spot is None:
```

```
        Print("Parking Lot Full: No spots available.")
```

```
        Return False
```

```
    R, c = spot
```

```
    Self.grid[r][c] = 'X'
```

```
    Print(f"Vehicle parked at spot ({r}, {c}).")
```

Return True

Def release_spot(self, row, col):

"""Release the spot back to vacant."""

If 0 <= row < self.rows and 0 <= col < self.cols:

If self.grid[row][col] == 'X':

Self.grid[row][col] = '.'

Print(f"Spot ({row}, {col}) released and now vacant.")

Return True

Else:

Print(f"Spot ({row}, {col}) is already vacant.")

Return False

Else:

Print("Invalid spot coordinates.")

Return False

Def main():

Lot = ParkingLot()

Print("Welcome to the Virtual Parking Assistant (ASCII Grid).")

While True:

Lot.display()

Print("Options:")

Print(" 1 – Park a vehicle")

Print(" 2 – Release a spot")

Print(" 3 – Exit")

Choice = input("Enter your choice: ").strip()

If choice == '1':

Lot.park_vehicle()

Elif choice == '2':

Try:

R = int(input("Enter row to release: "))

C = int(input("Enter column to release: "))

Lot.release_spot(r, c)

Except ValueError:

Print("Invalid input. Please enter numeric row and column.")

Elif choice == '3':

Print("Exiting Virtual Parking Assistant. Goodbye!")

Break

Else:

Print("Invalid choice. Please select a valid option.")

```
If __name__ == "__main__":
```

```
    Main()
```