Task# 01

Autonomous Dept.
Formulaelectric Racing-NUST

PYTHON PROGRAMMING:

(2 D Lists)

- In this task, the autonomous car detects blue and yellow cones marking the left and right track boundaries. Each frame gives a pair [left_cone_x, right_cone_x] (in meters), where a missing cone is represented as 0. The input data is:

```
cone_data = [
  [2, 8],
  [3, 0],
  [0, 9],
  [0, 0],
  [4, 10],
  [0, 0],
  [5, 11] ]
```

The goal is to calculate the lane's midpoint for every frame, even when cone data is missing. If both cones are detected, the midpoint is (left + right) / 2. If only one cone is detected, assume the missing cone's position — if only left is given, take right = left + 6, and if only right is given, take left = right - 6. When both cones are missing, find the previous and next valid midpoints and take their average to interpolate the missing value. Finally, loop through the list, apply these rules, and print the resulting list of midpoints that represents the estimated center path of the car.

- In this task, an autonomous car uses a **2D sensor grid** to measure distances (in cm) from nearby obstacles, where smaller values mean closer objects and 0 means no detection.

```
obstacle_map = [
  [0, 25, 15, 0],
  [10, 5, 0, 20],
  [0, 0, 30, 40],
  [8, 12, 0, 0]
]
```

Your program must loop through this grid and create a new 2D list called status_map, replacing each value based on distance:
≤10 → "DANGER", 11–25 → "CAUTION", >25 → "CLEAR", and 0 → "NO DATA".
Finally, print the status_map and count how many of each type appear to show the car's surrounding safety zones.