Question 1 – Map Loading and Publishing (Mandatory)

1. Given a 2D occupancy grid map provided as a .pgm file with a corresponding .yaml file:
    1. Write a ROS 2 node that reads the map files from disk.
    2. Convert the map into a nav_msgs/OccupancyGrid message.
    3. Publish the map on the /map topic.
    4. Ensure the map resolution, origin, and occupancy values are correctly set.

2. Visualize the published map in RViz.

Question 2 – Global Localization Using Particle Filter (Mandatory)

1. Implement a **particle filter–based global localization** system for a differential-drive robot operating in a known 2D map.

2. Use the following ROS 2 topics as inputs:
    1. /ekf/odom (nav_msgs/Odometry)
    2. /scan (sensor_msgs/LaserScan)
    3. /map (nav_msgs/OccupancyGrid)

3. Your particle filter must:
    1. Represent the robot state as **(x, y, θ)**.
    2. Initialize particles uniformly over free space in the map.
    3. Perform motion updates using odometry.
    4. Perform measurement updates using 2D LiDAR data.
    5. Apply a resampling strategy to maintain particle diversity.

4. Publish:
    1. The estimated robot pose as geometry_msgs/PoseWithCovarianceStamped.
    2. The TF transform from map to odom.

5. Visualize the particle distribution and estimated pose in RViz.

Question 3 – A* Path Planning on a 2D Map (Mandatory)

1. Using the occupancy grid map and the robot's current estimated pose:
    1. Implement the **A*** algorithm for 2D grid-based path planning.
    2. Accept a goal pose provided by the user.

2. The planner must:
    1. Avoid occupied cells.
    2. Use the map resolution and origin correctly.

3. Publish the computed path as a nav_msgs/Path message.

4. Visualize the planned path in RViz.

Question 4 – Simultaneous Localization and Mapping (SLAM) (Bonus)

Implement a **custom 2D SLAM system** to generate an occupancy grid map of the environment using the simulated differential-drive robot.

1. Design and implement a SLAM pipeline **from scratch**, without using existing SLAM or mapping packages (e.g., slam_toolbox, gmapping, Cartographer, or similar).

2. Use the robot's available sensors and data streams (e.g., odometry and 2D LiDAR) to:
    - Estimate the robot's pose over time.
    - Incrementally build a 2D occupancy grid map of the environment.

3. Publish the generated map as a nav_msgs/OccupancyGrid.

4. Save the final map to disk as:
    - a .pgm file, and
    - a corresponding .yaml file containing the map metadata.

5. Briefly describe the limitations of your SLAM implementation