Problem Statement

Develop a program that reads a Schedule and a Room Plan, both containing components represented by icons. Your task is to use the Schedule as a lookup table to identify and count the matching components present in the Room Plan. The expected result is a count of each component from the Schedule found in the Room Plan.

Instructions:

1. Data Files:
    - You will be provided with two sample data files:
        - Schedule File: Contains a list of components represented solely by icons DataSet
        - Room Plan File: Contains a detailed layout with various components, also represented by icons. DataSet
2. Task Overview:
    - Parse the Schedule: Extract the list of components (icons) from the Schedule file.
    - Analyze the Room Plan: Search for these components within the Room Plan file.
    - Count Matches: For each component from the Schedule, count how many times it appears in the Room Plan.
    - Result Compilation: Generate a report that lists each component from the Schedule alongside its corresponding count from the Room Plan.
3. Programming Requirements:
    - Language: You may use any programming language you are comfortable with.
    - Libraries and Tools: You are allowed to use external libraries or tools for image processing and pattern recognition if necessary.
    - Efficiency: Optimize your program for accuracy and performance.
4. Deliverables:
    - Source Code: Submit all source code files used in your program.
    - Instructions: Provide clear instructions on how to set up and run your program.
    - Report: Include a brief report (1-2 pages) detailing:
        - Your approach and methodology.
        - Any assumptions made.
        - Challenges encountered and how you addressed them.
        - Explanation of how your code works.
    - Output: Submit the result generated by your program using the sample data provided.

Skill Based Hiring

Evaluation Criteria:

- Accuracy: Correctly counts and reports the number of matching components.
- Efficiency: The program runs efficiently without unnecessary computational overhead.
- Code Quality: Clean, well-documented, and maintainable code following best practices.
- Problem-Solving Approach: Demonstrates a clear and logical methodology.
- Documentation: Clarity and completeness of the report and instructions.

Submission Guidelines:

- Deadline: 10/23/2024 (23rd October before 10 AM CST)
- Format: Compress all your files into a single ZIP archive named [YourName]_Assessment.zip
- Submission Method: Email the ZIP file to sameer@cognitive-sprints.in with the subject line "Computer Vision Assessment Submission - [Your Name]".