Laboratory 3: Password Guessing Program

Objective:

To create a Python program that can systematically guess a 4-digit password where each digit is between 1 and 5, without using any built-in modules or imports. This lab will strengthen your understanding of loops, string manipulation, and brute-force logic in Python.

Task:

Write a Python program that attempts to guess a 4-digit password using only the numbers 1, 2, 3, 4, and 5. The program should systematically try all possible combinations until it finds the correct password.

Requirements:

1. The password to be guessed should be a 4-digit string, where each character is one of the numbers between 1 and 5 (e.g., "1534").
2. **No imports or external libraries** are allowed.
3. Use nested loops to generate each possible combination of 4 digits.
4. Print each combination as the program attempts it.
5. When the correct password is guessed, print "Password found: " followed by the correct password and stop the program.

Instructions:

1. **Define the Password**: Hardcode the password in your program as a string (e.g., `password = "1534"`). This will be the password that the program attempts to guess.
2. **Set Up Possible Digits**: Define a list of possible digits, `['1', '2', '3', '4', '5']`, that the program will use to generate guesses.
3. **Generate Guesses**: Use four nested loops, each representing one digit of the password. Each loop should iterate through the list of possible digits.
4. **Form Each Guess**: For each combination generated by the loops, form a 4-digit string representing a password guess (e.g., `guess = i + j + k + l`).
5. **Check the Guess**: If the guess matches the password, print "Password found: " followed by the password, and exit the program.
6. **Output**: Print each attempted combination for reference as the program tries each guess.