WEEK 6

1. Your task is to create a simple password-checking mechanism in C using different types of loops (for, while, and do-while). The goal is to allow a user to enter a password up to five times and check whether the entered password is correct. The password in this scenario is an **integer**.

Here's the breakdown of the requirements:

Password Checking Function: You need to write a function called checkPasswordWithFor, checkPasswordWithWhile, and checkPasswordWithDoWhile that takes an integer as an argument and returns a boolean indicating whether the entered password matches the predefined correct password.

Limiting Attempts: The user is allowed a maximum of five attempts to enter the correct password.

User Input: In each attempt, prompt the user to enter the password.

Validation: After each entry, the program should check if the entered password is correct.

If the entered password is correct, the program should display "Correct" and stop asking for more input.
If the entered password is incorrect, the program should prompt the user to try again, unless the maximum number of attempts, which is 5, has been reached.
If the user fails to enter the correct password after five attempts, the program should display "Fail".

2. Write a C program to calculate the Greatest Common Divisor (GCD) of two given positive integers. Although there are efficient algorithms available for solving the GCD problem, we aim to implement a straightforward approach. First, determine the smaller of the two numbers. Then, initiate a loop that counts up from 1 to this smaller number, which is the largest possible GCD. In each iteration, divide both original numbers by the current number in the loop. If both divisions yield a remainder of 0, then you have found a nominee for the GCD. In the worst-case scenario, the loop will end up finding any common divisor except 1, which is the GCD by definition for any pair of integers.