Question 1:

Write a C program that uses `fork` to create a child process. The child process should display its process ID (PID), and the parent process should display the PID of the child.

Question 2:

Create a C program that simulates a simple bank account system. Use `fork` to create child processes representing different account holders. Implement deposit and withdrawal operations with proper synchronization.

Question 3:

Develop a C program that creates a child process to find and print all prime numbers in a given range. The parent process should specify the range, and the child process should perform the computation.

Question 4:

Write a C program that uses `fork` to create child processes that sort different sections of a large array concurrently. After sorting, merge the sorted sections in the parent process to obtain a fully sorted array.

Question 5:

Create a C program that utilizes `fork` to create a child process that calculates the square of a given number using a separate executable file. The child process should execute this external program and return the result to the parent.

Question 6:

Implement a C program that models a restaurant with multiple cooks and customers. Use `fork` to create child processes representing cooks and customers, and ensure proper synchronization and communication between them.