

## Part 1: Multithreading (Dividing Tasks Among Threads)

### Problem 1: Sum of an Array Using Multiple Threads

- Write a C program that calculates the sum of a large integer array using multiple threads.
- Divide the array into equal chunks, assign each chunk to a separate thread, and compute partial sums.
- The main thread should sum up the partial sums to get the final result.

### Problem 2: Matrix Multiplication Using Threads

- Implement matrix multiplication using multiple threads.
- Each thread should be responsible for computing a single row or a single element of the resulting matrix.

### Problem 3: Finding Maximum Element in an Array

- Create a multithreaded program where multiple threads each find the maximum in a portion of the array.
- The main thread then finds the maximum among the partial results.

### Problem 4: Word Count in a File Using Multiple Threads

- Given a large text file, count the occurrences of a specific word using multiple threads.
  - Divide the file into chunks, process each chunk in a separate thread, and aggregate results.
- 

## Part 2: Synchronization (Using Mutex and Semaphores)

### Problem 5: Counter with Mutex (Race Condition Handling)

- Write a program where multiple threads increment a shared counter.
- Use a **mutex** to ensure correct synchronization.

### Problem 6: Producer-Consumer Problem Using Semaphores

- Implement the classic **Producer-Consumer** problem using a bounded buffer.
- Use semaphores for synchronization between producer and consumer threads.

### Problem 7: Reader-Writer Problem (Synchronization)

- Implement a **Reader-Writer** solution where:
  - Multiple readers can read simultaneously.
  - A writer must have exclusive access to write.
  - Use a **mutex** and **semaphores** for synchronization.

**Problem 8: Dining Philosophers Problem Using Mutex**

- Implement the **Dining Philosophers** problem using mutexes to avoid deadlock and starvation.

**Problem 9: Print Odd and Even Numbers Alternately**

- Write a program where two threads print numbers alternatively:
  - One prints **odd numbers**.
  - The other prints **even numbers**.
- Use a **mutex** or a **semaphore** for synchronization.