CPP-A23: Multithreading <thread>

Assignment 1: Sum of Array Using Threads

Divide an array into two halves and calculate their sum using two threads. Combine the results in the main thread.

Input:

```
Array: [1, 2, 3, 4, 5, 6]
```

Output:

```
Sum by Thread 1: 6
Sum by Thread 2: 15
Total Sum: 21
```

Assignment 2: Parallel Matrix Multiplication

Multiply two matrices in parallel, where each thread computes one row of the result.

Input:

```
Matrix A: 2×2
1 2
3 4

Matrix B: 2×2
5 6
7 8
```

Output:

```
Result Matrix:
19 22
43 50
```

Assignment 3: Bank Account Deposit and Withdrawal (With Mutex)

Simulate concurrent deposit and withdrawal operations on a shared account using mutex to avoid race conditions.

Input:

```
Initial Balance: 1000
Thread1 deposits 500
Thread2 withdraws 300
Thread3 withdraws 400
```

Output:

```
Final Balance: 800
```

Assignment 4: Producer-Consumer Problem (Buffer Synchronization)

Use a queue and mutex to implement the classic producer-consumer problem.

Input:

```
Producer produces 5 items: 10, 20, 30, 40, 50
Consumer consumes them
```

Output:

```
Produced: 10
Consumed: 10
Produced: 20
Consumed: 20
...
```

Assignment 5: Parallel Find Maximum in Array

Use multiple threads to find the maximum element in an array, with each thread working on a segment.

Input:

```
Array: [3, 12, 8, 25, 7, 17]
```

Output:

```
Thread 1 Max: 12
Thread 2 Max: 25
Global Max: 25
```

Assignment 6: Prime Number Counter Using Threads

Use two threads to count how many prime numbers are there between 1 and N.

Input:

```
Range: 1 to 20
```

Output:

```
Thread 1 Primes: 2, 3, 5, 7
Thread 2 Primes: 11, 13, 17, 19
Total Primes: 8
```

Assignment 7: File Copying with Threads

Create two threads, each copying half of a file into another file. Use mutex to synchronize writing.

Input:

```
Source File: "data.txt" (100 lines)
Destination File: "copy.txt"
```

Output:

```
Thread 1 copied lines 1-50
Thread 2 copied lines 51-100
File copied successfully.
```

Assignment 8: Chat Simulation Using Threads

Simulate a chat between two users using threads, where each thread prints messages alternately.

Input:

```
User1 messages: ["Hi", "How are you?"]
User2 messages: ["Hello", "I'm fine"]
```

Output:

```
User1: Hi
User2: Hello
User1: How are you?
User2: I'm fine
```

Assignment 9: Timer Countdown and Notification

One thread acts as a countdown timer, while the other shows a notification when time is up.

Input:

```
Countdown from: 5 seconds
```

Output:

```
5... 4... 3... 2... 1...
Time's Up! (from notification thread)
```

Assignment 10: Multi-threaded Sorting (Odd-Even Sort)

Implement **odd-even sort** using multiple threads.

Input:

```
Array: [5, 3, 8, 4, 2, 6, 1, 7]
```

Output:

```
Sorted Array: [1, 2, 3, 4, 5, 6, 7, 8]
```

Happy Coding!