1. Write a Java program that uses two threads to print numbers from 1 to 100. One thread should print odd numbers, and the other should print even numbers. Ensure proper synchronization to alternate between the two threads.
2. Write a Java program that uses two threads to increment and decrement a shared counter variable concurrently. Implement proper synchronization to avoid race conditions and ensure that the final value of the counter reflects the correct total changes made by both threads.
3. Develop a Java program that generates a simple Java thread, which prints "Hello, World!" upon execution.
4. Create a Java program that utilizes multiple threads to sort an array of integers.
5. Implement a Java program for matrix multiplication using multiple threads.
6. Develop a Java program that employs multiple threads to calculate the sum of all prime numbers up to a specified limit.
7. Write a Java program that creates two threads to find and print even and odd numbers from 1 to 20.
8. Write a Java program to print brackets synchronously like (,{,[,],},)or [{[({[]})]}] using 3 threads each for one type of brackets.
9. Design a Java program to calculate the sum of all prime numbers up to a given limit using multiple threads. Each thread will contribute to the calculation by checking a specific range of numbers for primality and adding the prime numbers found to the overall sum. Additionally, when a thread finds a prime number, it should print the prime number along with the thread name.