Task 6: Executors, Concurrent Collections, CompletableFuture
Use an ExecutorService to parallelize a task that calculates prime numbers up to a given number and then use CompletableFuture to write the results to a file asynchronously.
ANS:

```
package com.Day23;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.CompletableFuture;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.TimeUnit;
public class Task6 {
  public static void main(String[] args) {
    int maxNumber = 1000; // Maximum number up to which prime numbers will be calculated
    ExecutorService executor = Executors.newFixedThreadPool(4); // Create a fixed-size thread pool
with 4 threads
    CompletableFuture<List<Integer>> primeFuture = CompletableFuture.supplyAsync(() -> {
      return calculatePrimes(maxNumber); // Asynchronously calculate prime numbers
    }, executor);
    CompletableFuture<Void> writeToFileFuture = primeFuture.thenAcceptAsync(primes -> {
      try {
        writePrimesToFile(primes, "primes.txt"); // Asynchronously write prime numbers to a file
      } catch (IOException e) {
        e.printStackTrace();
      }
    }, executor);
    // Shutdown the executor after all tasks are completed
    CompletableFuture<Void> allTasksFuture = CompletableFuture.allOf(primeFuture,
writeToFileFuture);
    allTasksFuture.thenRun(() -> {
      executor.shutdown();
      try {
        executor.awaitTermination(5, TimeUnit.SECONDS);
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    });
  }
  public static List<Integer> calculatePrimes(int maxNumber) {
    List<Integer> primes = new ArrayList<>();
    for (int num = 2; num <= maxNumber; num++) {
      boolean isPrime = true;
      for (int i = 2; i <= Math.sqrt(num); i++) {
        if (num % i == 0) {
          isPrime = false;
          break;
        }
      }
      if (isPrime) {
```

```
                primes.add(num);
            }
        }
        return primes;
    }
    public static void writePrimesToFile(List<Integer> primes, String fileName) throws IOException {
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName))) {
            for (Integer prime : primes) {
                writer.write(prime.toString());
                writer.newLine();
            }
        }
        System.out.println("Prime numbers written to file: " + fileName);
    }
}
```