5CS021 – Numerical Methods and

Concurrency **LAB REPORT – Week10**

Attempt all questions

1. Write a multithreaded C program with three threads that increment a shared counter. Use two global variables: one for the counter and one for a check variable that stores which thread first reaches 50. Implement three functions: `count_thread1()`, `count_thread2()` and `count_thread3()`, where each thread increments the counter. When a thread reaches 50, it updates the check variable with its thread number (1, 2 or 3) and exits. Ensure proper synchronization to avoid race conditions when modifying the counter.The main function should print which thread first reached 50, based on the check variable.

2. Write a multithreaded C program where the user is prompted to enter a positive integer N. The program should create two threads, and both threads will call the same function, `add_natural()`, to compute the sum of the first N natural numbers. The shared variables `sum` (initialized to 0) and `i` (initialized to 1) should be used globally. Each thread will alternately add numbers to `sum` from 1 to N, and proper synchronization should be applied to avoid race conditions when updating these shared variables. After both threads have finished executing, the main function should print the total sum and indicate which thread added the final number to the `sum`. The program should ensure that each thread adds numbers one by one, and the last thread to add a number should be noted.