GRIFFITH COLLEGE DUBLIN

COMPUTING ASSIGNMENT TITLE SHEET

Course:	BSCH Computing
Stage/Year:	4
Module:	Distributed Systems
Semester:	Semester I
Assignment Number:	Assignment 2
Date of Title Issue:	Dec 9th, 2020
Assignment Deadline:	Jan 4th, 2021 - 11:55pm
Assignment Submission:	Moodle Upload
Assignment Weighting:	20/40

Standard penalties will be applied to work that is submitted late, as per faculty guidelines.

All work ***must be your own***
If you copy from someone else, both parties will be awarded a grade of 0.

Learning Outcomes

Programme and related module learning outcomes that this assignment is assessing:

1,3

Assessment Criteria

Assessment criteria applied to this assignment, such as:

- □ Presentation
- □ Code Structure and Cleanliness
- □ Code Performance
- □ Appropriate Output
- □ Appropriate Method Selection

Assignment 2: Primes

Ensure your code is well commented, as well as neat and readable. Code that fails to compile will incur a **penalty of 30%.**

Work that is submitted late will incur standard penalties as per faculty guidelines.

Print your name and student number to the console for all programs you write.

You are given a method, *primeCounter()* which accepts an integer *n* and returns an integer containing the amount of prime numbers in n.

Part 1 - 50%

```
int primeCounter(int n) {
     int count=0;
    for (int a=2 ; a < n ; a++) {
        bool isPrime = true;
        for (int c=2 ; c*c <= a ; c++) {
            if(a % c == 0) {
                isPrime = false;
                break;
            }
        }
        if(isPrime) count++;
    }
     return count;
}
```

Write an MPI program using 4 nodes. You are required to find out how long each node takes to run the *primeCounter()* method where *n* is the following number:

> For each node the value passed to primeCounter n is $25^{(rank+2)}$
> *(Twenty five to the power of the rank of each process plus two)*

Only rank 2 should output the duration results from all nodes.

Choose an appropriate MPI function to help with the gathering (as only node 2 will output duration values from all nodes - Send/Receive is a nasty solution!).

Part 2 - 50%

The *primeCounter()* function given is a sequential method that returns the amount of prime numbers in a given integer *n.* Create a second MPI program, which utilises a new *primeCounter()* function that is optimised to utilise MPI instead of running sequentially.

The new *primeCounter()* function should take an array of ints and its size as input and return the total amount of prime numbers contained in the array. It should utilise the appropriate MPI functions to divide the work between nodes, gather the multiple results and return a single count.

Each node should print the amount of primes it has encountered, and only one should print out the total count of primes.

Test this new function using 8 nodes and an array of ints from 1-3999. Print out the total amount of prime numbers encountered in this array.

Extra marks will be given for added code efficiency if your function is able to discard approximately half of the non-primes instantly.

Ensure your code contains a comment at the beginning that shows the following:

> *Your name.*
>
> *Your student number.*
>
> *Your processor. (i7 6700k, i5 5500, Ryzen 5 2200G, etc...)*
>
> *Your output showing the four duration values for part I. (The console logging output that rank 2 spits out after all calculations).*

For example:

```
/*
    Name:   Jennifer Lebron
    No:     2991234
    uP:     i7 6700k
    Result: 1 2 3 4
*/
```

Submit your two .cpp files (you may add makefiles if you choose) archived into a **.zip** file.

Tips:

MPI_Wtime() is your friend.

Read the brief, then read it again. Then read the brief.

You will lose marks for untidy code.

You will lose marks for lack of appropriate comments.

You will lose marks if your code does not compile and run.

You will lose marks for confusing console output/not outputting information.

You will lose marks for not using the most appropriate MPI functions.

You will be graded on the quality of your code.