GRIFFITH COLLEGE DUBLIN

COMPUTING ASSIGNMENT TITLE SHEET

Course:	BSCH Computing
Stage/Year:	4
Module:	Distributed Systems
Semester:	Semester II
Assignment Number:	Assignment 3
Date of Title Issue:	November 24th, 2021
Assignment Deadline:	December 12th, 2021 - 11:55pm
Assignment Submission:	Moodle Upload
Assignment Weighting:	20%

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 3: Parallel Matrix Multiplication

Introduction

In this assignment you will be tasked with building a fully working parallel matrix multiplier. Matrix multiplication is a common task in many scientific applications and large matrices take time to compute. However, if divided in the right way the task can be parallelised efficiently.

In this assignment you will need to multiply two matrices together using the stripe method of matrix multiplication. The block method is more efficient but it is recommended that you implement the stripe first before you try the block. You will have three matrices A,B and C that represent the operation A x B = C. You will be required to determine how many nodes are in your compute group and create even stripes for A to be divided amongst the nodes. All nodes will receive a copy of B. Each node will do a matrix multiplication of its stripe of A with the matrix B to generate its stripe of C. Finally all nodes will use the gather operation to send back the stripes to the coordinator node wherein it will be reassembled into a single matrix.

You may assume that all of your matrices are square (NxN) and that N is evenly divisible by the number of nodes. The support file you will receive for generating matrices will generate 8x8 matrices and thus will work well with 4 nodes for testing. All matrices must be passed in on the command line along with the number of elements in a row or column e.g. to pass in two matrices you would do something like this:

```
mpirun -n 4 ./assignment03 matA.dat matB.dat 8
```

The matrices should only be read by the coordinator process.

Submission:

Submit **one .cpp** file containing your MPI source code (do not include header or make files). Any format outside of this will incur a 10% penalty.

For the purposes of this assignment you will only need four standard header files <iostream>, <cstdlib>, <cmath> and <mpi.h>. Do not submit any additional header files.

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.

Task List:

01) write a main method that will initialise MPI, figure out the world rank and world size. Rank 0 should be the coordinator while all other ranks should be participants. Then finalise MPI and return a status of 0 to the OS (5%)

02) Add the following helper methods to your code (15%)

- *printMatrix()* will print a 2D matrix to the console
- *dotProduct()* will that multiply a row of matrix A with a column from matrix B that will return a single value that is the dot product of the row and column
- *multiplyStripe()* takes in a stripe of A, a matrix of B and computes a stripe of C.

03) Write coordinator and participant methods that perform the following interactions (80%)

- read in matrices A and B from disk. (coordinator only)
- broadcast a message stating that the computation will be performed if the matrices are present and correct and that we have the correct number of command line arguments. Give a different message otherwise.
- Take part in multiple broadcasts that will tell all nodes the size of a matrix, the size of a stripe and the size of an individual row.
- Allocate the necessary memory for the stripes and matrices required.
- Take part in a scatter to distribute the stripes of A and take part in a broadcast to get the full copy of B.
- perform the multiplication and take part in a gather to send all stripes of C back to the coordinator.
- print out the matrix (coordinator only) and deallocate all memory.

Tips:

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.

You may want to check your results using a Matrix Multiplication Calculator:
https://matrix.reshish.com/multiplication.php