



## Griffith College Dublin

**Course:** Parallel and Distributed Programming

**Module:** PDP

**Semester:** Semester II

**Assignment Number: 2 OpenMP and MPI**

**Date of Title Issue:** 12th. April 2021

**Assignment Deadline:** 9th. May 2021

**Assignment Weighting:** 20/50

**Please state the assignment title / brief. Please specify details such as:** Answer the questions on the accompanying sheet.

### Learning Outcomes

**Please state the programme and related module learning outcomes that this assignment is assessing.** 1,2,4, 5,6

### Assessment Criteria

Please state the assessment criteria applied to this assignment, such as:

- Correctness of the work.
- Presentation, including compliance with the specified file format.
- Evidence of critical thinking and analysis.
- Originality, quality and thoroughness of the work.
- Research correct academic approach.
- Proper treatment of sources.
- Demo of code will be in lab, failing to show work in lab receive deduction of 50% of total assignment grade.

*Academic Dishonesty: All of your assignments need to represent your own effort. Assignments should be done without consultation with other students and you should not share your source code with others. Any assignment submitted that is essentially the same, as someone else's will not be accepted. **ALL matching assignments will receive 0 credits.***

Your task is to search for a number in a 2-D matrix of random single digit numbers, the idea of the solution is to parallelize the search between all possible threads and count each occurrence.

Make sure to include timing to your code, you need to submit two solutions using MPI and OpenMP:

1. **OpenMP C file and results of execution.** **30%**
2. **Find neighbouring numbers.** **20%**
3. **Repeat part 1 using MPI C file and results of execution.** **50%**

Example: looking for digit 10

1	3	4	7	3	9	6	9	7	2
3	5	0	4	1	9	9	9	0	1
5	1	2	7	9	4	2	7	9	1
7	5	8	9	2	7	6	4	2	3
9	0	9	6	3	2	0	2	3	8
1	4	6	2	1	4	8	5	6	1
0	3	0	4	7	2	9	1	0	4
4	1	2	3	1	0	3	7	5	8
2	3	5	0	8	5	9	8	1	6
8	6	4	8	6	2	6	0	7	2