

## Week 8 - Assignment (3 marks)

**Note: Try to make the best use of appropriate C++ features.**

### Objective:

- Implement a custom, robust matrix operation program.
- Apply multithreading (`std::thread`) to enhance computational efficiency for large matrix operation.
- Compare the performance of single-threaded and multithreaded computations.

### Tasks:

A custom matrix operation is defined as follows:

Given two matrices, where  $A$  is an  $m \times n$  matrix and  $B$  is an  $n \times p$  matrix, their result  $C$  is an  $m \times p$  matrix, with each element  $C[i][j]$  computed according to the following steps:

1. Compute the dot product of the  $i$ -th row of  $A$  and the  $j$ -th column of  $B$ , defined as:

$$\text{dot\_product} = \sum_{k=1}^n A[i][k] \times B[k][j]$$

2. If the remainder of the dot product divided by 100 is greater than 50, set  $C[i][j] = 1$ .
3. Otherwise, set  $C[i][j] = 0$ .

Then,

1. Implement the matrix operation using a single-threaded approach.

`customOpSingle`

2. Implement the matrix operation using a multithreaded approach.

`customOpMulti`

### Input:

Please use **the provided `run_wa8.cpp`** to test your implementation.

Don't modify the provided code!

### Submit:

- 1, **all C++ source code**
- 2, **wa8.txt**: a txt file contains all the source code for *plagiarism review*.
- 3, **output.jpg** (or png, bmp): a screenshot of the output by your program.

**Please refer to the submission page for the Marking Rubric.**