**QUESTION SHEET: PARALLEL SQUARE DETECTOR**

**Introduction**

You are expected to develop a parallel program using MPI and OpenMP in C or C++ language. The program is expected to:

1. Detect how many squares (for digit ‘2’) are available in a given binary-matrix.
2. Identify the coordinates of top-left and bottom-right corner of each square.

As an example, a binary-matrix given as shown in **Figure 1**. The matrix’s dimension is 20 x 30. It contains 3 squares as outlines by the red boxes. Example of program output is shown in **Figure 2**.

A picture containing diagram

Description automatically generated

**Figure 1**: Example of a 20 x 30 matrix

Graphical user interface, text, application

Description automatically generated

**Figure 2**: Example output with input data as shown in Fig. 1.

You are required to produce a parallel program using C or C++ that will perform the following operations:

1. The program will read a binary-matrix in text file named “*exam-data.txt*”. The file containing 5000 x 5000 matrix of digit 0, 1 and 2 elements. The *exam-data.txt* is provided in exam-data.zip together with the md5sum.txt
2. As explained in Introduction section above, your parallel program is expected to: i. Detect how many squares are available in a given binary-matrix (for

digit ‘2’).

ii. Identify the coordinates of top-left and bottom-right corner of each square.

1. In your program, you MUST implement at least MPI, OpenMP, MPI Point to Point Communication, MPI Collective Communication, Decomposition.
2. The algorithm MUST be scalable. It means that is supports flexible number of MPI processes entered by users when execute the program.

Note:

Even though the matrix is represented as 5000 x 5000 2-dimensional array. Your matrix buffer in the program may just use 1-dimentional array as it is more practical with the MPI functions.