School of Computing

UNIVERSITI TEKNOLOGI MALAYSIA

RESEARCH UNIVERSITY

FINAL ASSESSTMENT SEMESTER II 2020/2021

COURSE CODE	**:**	**SCSR3223**
COURSE TITLE	**:**	**HIGH PERFORMANCE AND PARALLEL COMPUTING**
PROGRAM	**:**	**SCSR**
TOTAL TIME	**:**	**2 WEEKS**
DATE	**:**	**27 / 06 / 2021 – 10 / 07 /2021**

(GENERAL INSTRUCTION):

Submission is done by uploading the files containing your answer to the e-learning. The answers should contain the following:

- **Task 1 deliverable: The source code (.c or .cpp)**
- **Task 2 deliverable: The report (PDF format)**
- **Task 3 deliverable: The video (MP4 or AVI format)**

Note: Please do not ZIP your files. Upload them as separate files.

This test will contribute 30% towards the total marks of 100%.

Warning!

Students who are caught cheating on the examination will be reported to the disciplinary board for possible suspension of the student for one or two semesters.

Name	
Metric No	
Year / Course	
Section (Circle)	**01 / 02**
Lecturer	**Nur Haliza Binti Abdul Wahab**

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:

a. Detect how many squares (for digit '2') are available in a given binary-matrix.
b. 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**.

```
0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2
1 2 2 2 1 1 2 0 1 1 0 1 1 0 1 2 0 1 2 0 1 1 0 1 0 0 1 0 0 2
1 2 2 2 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2
1 2 2 2 1 0 0 1 1 0 1 2 0 1 0 0 1 0 0 1 1 0 1 1 0 1 1 0 1 0
0 1 2 0 1 0 1 0 0 1 2 0 1 0 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 1
1 2 0 2 1 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 2
1 0 0 2 1 2 0 1 1 0 1 2 0 1 2 0 1 0 2 2 2 2 2 2 0 1 1 0 1 2
1 2 0 1 1 0 0 1 2 0 1 2 0 1 1 0 1 2 2 2 2 2 2 2 0 1 1 0 1 0
1 2 1 2 1 2 0 1 1 0 1 2 0 1 2 0 1 1 2 2 2 2 2 2 0 1 0 0 1 2
1 1 2 0 1 0 0 1 1 0 1 2 0 1 1 0 1 1 2 2 2 2 2 2 0 1 0 0 1 0
1 2 2 2 1 2 0 1 2 0 1 2 0 1 0 0 1 0 2 2 2 2 2 2 0 1 0 0 1 2
1 1 2 0 1 0 0 1 0 0 1 2 0 1 1 0 1 0 2 2 2 2 2 2 0 1 1 0 1 0
1 2 0 2 1 2 0 1 0 0 1 2 0 1 2 0 1 0 2 0 2 0 2 0 1 1 0 1 2 1
1 1 2 0 1 0 0 1 2 0 1 1 0 1 0 0 1 1 1 1 0 0 1 2 0 1 0 0 1 0
1 2 0 1 1 2 0 1 0 0 1 1 0 1 0 0 1 1 2 2 1 2 0 0 0 1 2 0 1 2
1 2 1 0 1 0 0 1 2 0 1 0 0 1 0 0 1 0 2 2 2 0 1 2 0 1 1 0 1 0
1 2 0 2 1 2 0 1 1 0 1 0 0 1 0 0 1 0 1 0 1 2 0 1 0 1 0 0 1 2
1 0 0 1 1 0 0 1 1 0 1 2 0 1 1 0 1 2 0 0 2 1 1 0 0 1 1 0 1 0
1 2 2 1 1 2 0 1 2 0 1 1 0 1 1 0 1 1 0 1 0 2 1 2 0 1 0 0 1 2
1 2 0 2 1 0 0 1 2 0 1 2 0 1 2 0 1 2 1 1 0 2 0 0 0 1 2 0 1 0
```

Figure 1: Example of a 20 x 30 matrix

```
Number of square: 3

Square 1 coordinates [top-left corner]: (2, 2)
Square 1 coordinates [bottom-right corner]: (4, 4)

Square 2 coordinates [top-left corner]: (19, 7)
Square 2 coordinates [bottom-right corner]: (24, 12)

Square 3 coordinates [top-left corner]: (19, 15)
Square 3 coordinates [bottom-right corner]: (20, 16)
```

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

QUESTION 1 [50 MARKS]

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

a. 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

b. 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.

c. In your program, you MUST implement at least MPI, OpenMP, MPI Point to Point Communication, MPI Collective Communication, Decomposition.

d. 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.

QUESTION 2 [25 MARKS]

You are required to write a report regarding the solution you proposed in (Question 1). In the report should have the following:

a. Provide a simple guide or steps how to compile and execute your program.

b. Briefly discuss the model of parallel algorithm applied in the solution.

c. Explain, how do you implement decomposition in designing your algorithm?

d. Explain
   i. Which part of your solution implements MPI multi-processes computations?
   ii. What are tasks assigned to those processes?

e. Explains:
   i. Which part of your solution implements OpenMP multi-tread computations?
   ii. What are tasks assigned to those threads?

f. Explain MPI communication functions used in the solutions and its purpose.

g. Provide the source codes and example of program output in the Appendix.

QUESTION 3 [25 MARKS]

Record a short video of your demonstration of the program (outcome of Question 1) and present the outcome of Question 2. Duration of the video is not more than 10 minutes. Your video should include your face while presenting (as shown in example **Figure 3**). In the beginning of the video, you should have a slide containing your details (name, matric number and your Academic Program).

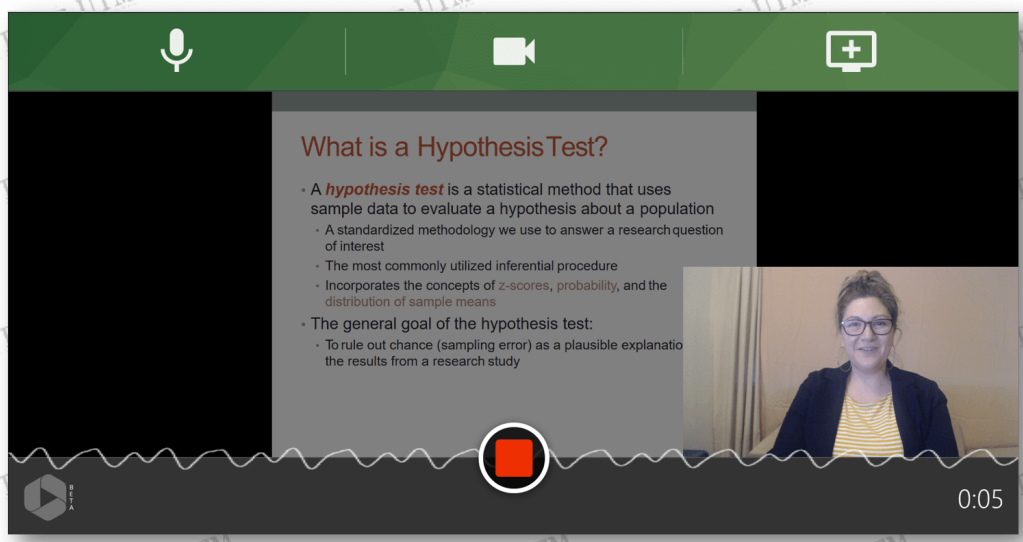

Figure 3: Example of presentation session with presenter's video thumbnail.

= END OF QUESTION =