[This question paper contains 4 printed pages.]

Your Roll No.....

Sr. No. of Question Paper : 1764 **L**

Unique Paper Code : 2342013603

Name of the Paper : Introduction to Parallel Programming (DSC)

Name of the Course : **B.Sc. (Hons.) Computer Science**

Semester : VI

Duration : 3 Hours Maximum Marks : 90

Instructions for Candidates

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. **SECTION A** is compulsory.
3. Attempt **any 4** questions from **SECTION B**.
4. Parts of a question must be answered together.

SECTION A

1. (a) What is parallel computing? Mention any two applications of parallel computing. [3]

   (b) How are tasks created and synchronized? Support your answer with a diagram. [3]

   (c) Consider two processes P0 and P1 executing the following MPI code using blocking non-buffered MPI communication: [4]

```
P0                              P1
send(&x, 1, 1);                 send(&y, 1, 0);
receive(&y, 1, 1);              receive(&x,1, 0);
```

   Explain whether a deadlock will occur. Justify your answer.

   (d) Differentiate between `MPI_Gather` and `MPI_Gatherv`. When would you prefer `MPI_Gatherv` over `MPI_Gather`? Justify your answer. [4]

(e) What are threads in parallel computing? Rewrite the following code using OpenMP directives to enable parallel execution across multiple processors. [4]

```
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        sum[i][j] = func(i, j);
```

(f) Define vertical and horizontal waste in superscalar architecture with an example. [4]

(g) Define true data dependence in superscalar execution. Illustrate your answer with example. [4]

(h) Differentiate between SIMD and MIMD architectures with a suitable example. [4]

SECTION B

2. (a) Explain the logical memory model of threads with the help of a diagram. Discuss any four advantages of threaded programming models over message-passing models. [7]

(b) Consider multiplying two matrices A and B of size 12 × 12 using the block matrix multiplication algorithm with a block size of 4 × 4. Answer the following: [8]

i. How many blocks will each matrix be divided into?
ii. How many block multiplications are required to compute one block of the result matrix?
iii. How many scalar operations (multiplications and additions) are required for one block multiplication?
iv. Calculate the total number of scalar operations required for the complete matrix multiplication.

3. (a) Explain the UMA (Uniform Memory Access) and NUMA (Non-Uniform Memory Access) architectures with the help of suitable diagrams. Compare the two architectures based on memory access time, scalability, and performance. [7]

(b) Consider the following OpenMP parallel code: [8]

```
int sum = 0;
#pragma omp parallel for
for (int i = 0; i < n; i++) {
    sum += a[i];
}
```

Answer the following:

i. What is a race condition in the context of parallel programming?

ii. Identify the race condition present in the above code, clearly stating which variable is affected and why.

iii. Explain why the race condition occurs in this code.

iv. Describe one method to prevent the race condition in the above code.

4. (a) Explain the steps involved in the Odd-Even Sort algorithm for parallel sorting. Apply the algorithm to sort the sequence [9, 3, 8, 2, 4] using 5 processors. [7]

(b) Explain the PRAM (Parallel Random Access Machine) model. Describe its different subclasses (EREW, CREW, CRCW and ERCW) based on concurrent memory access. [8]

5. (a) In a parallel application, processes exchange boundary data using blocking send and receive operations in MPI. [7]
Explain with example how improper ordering of communication can lead to deadlock. Describe any two communication strategies to avoid deadlock.

(b) Write pseudocode for row-wise parallel matrix multiplication of two $n \times n$ matrices A and B using the following MPI routines: [8]

i. `MPI_Comm_rank`

ii. `MPI_Scatter`

iii. `MPI_Bcast`

iv. `MPI_Gather`

6. (a) Consider a parallel program in which multiple threads update shared variables *sum* and *product*. Write pseudocode using a critical section and a reduction clause to ensure mutual exclusion. Which method is more efficient? Justify your answer. [7]

(b) Differentiate between the following pairs: [8]
(a) `MPI_MAX` and `MPI_MAXLOC`
(b) `MPI_Scatter` and `MPI_Scatterv`
(c) `MPI_Bcast` and `MPI_Gather`
(d) `MPI_Reduce` and `MPI_Allreduce`

7. (a) Explain Dijkstra's algorithm for single-source shortest paths. State its three main steps and write pseudocode illustrating how these steps can be implemented using collective communication operations to compute shortest paths from a given source vertex '*s*' to all other vertices. [7]

(b) A merge sort algorithm is parallelized using OpenMP task constructs, where recursive calls generate parallel tasks. However, the parallel version performs slower than the sequential version. [8]
Explain how *recursive task spawning* works in this context with a code example. List and explain five common pitfalls that may lead to this performance degradation.