Semester: VII	Department of Information Science and Engineering	AY: 2026-27
Course Code: BCS702	Course Name: PARALLEL COMPUTING	

Question Bank

Q.	Module-01	M	L	C
1.	Explain Flynn's classification of parallel computers. Differentiate between SIMD and MIMD systems with neat diagrams.	8	L2	CO1
2.	Explain the architecture of SIMD systems and discuss vector processors and graphics processing units (GPUs) as SIMD systems.	8	L2	CO1
3.	Explain shared-memory and distributed-memory MIMD systems. Compare the two with neat diagrams.	10	L2	CO1
4.	Describe any three interconnection networks used in shared-memory and distributed-memory systems. Compare direct and indirect interconnects.	10	L2	CO1
5.	What is the cache coherence problem? Explain snooping cache coherence and directory-based cache coherence.	10	L2	CO1
6.	Explain false sharing and its effect on performance in shared-memory systems with an example.	6	L2	CO1
7.	Discuss how processes and threads are coordinated in shared-memory and distributed-memory programs. Explain race conditions, critical sections and mutual exclusion.	8	L2	CO1
8.	Explain the need for parallel programming. Discuss why serial programs cannot simply be converted into parallel programs.	6	L2	CO1
9.	Explain UMA and NUMA shared-memory systems with neat diagrams.	8	L2	CO1
10.	Explain the crossbar, ring, toroidal mesh and hypercube interconnection networks. Define bisection width and bandwidth.	10	L2	CO1
	Module-02			
11.	Define speedup and efficiency of a parallel program. Explain the speedup and efficiency in MIMD systems with the help of an example.	8	L2	CO2
12.	State Amdahl's law. Derive the expression for speedup and discuss its limitations.	8	L2	CO2
13.	A serial program takes 20 seconds, of which 90% can be parallelized. Use Amdahl's law to find the speedup and efficiency for 4 and 8 processors.	5	L3	CO2
14.	Explain scalability in MIMD systems. Differentiate between strongly scalable and weakly scalable programs.	8	L2	CO2
15.	Explain how timings of MIMD programs are taken. Discuss the use of MPI_Wtime, barriers and reporting the maximum elapsed time across processes.	8	L2	CO2
16.	Explain the programming of hybrid systems and discuss GPU performance considerations.	8	L2	CO2

Aditya College of Engineering and Technology

17.	Discuss the performance issues in MIMD systems, namely parallel overhead, load imbalance and communication cost, and suggest ways to improve performance.	10	L3	CO2
18.	A parallel program takes 24, 13, 7.5 and 4.4 seconds on 1, 2, 4 and 8 processors respectively. Compute the speedup and efficiency for each case.	5	L3	CO2
19.	Explain how GPU performance is measured. Compare the performance of a GPU with that of a CPU for data-parallel programs.	8	L2	CO2
20.	Explain the effect of problem size on the scalability of a parallel program with an example.	6	L2	CO2
	Module-03			
21.	Explain the basic MPI functions MPI_Init, MPI_Finalize, MPI_Comm_size, MPI_Comm_rank, MPI_Send and MPI_Recv with their syntax. Write an MPI program in which each process sends a greeting to process 0.	10	L3	CO3
22.	Explain the trapezoidal rule. Develop an MPI program to implement the trapezoidal rule using point-to-point communication.	10	L3	CO3
23.	Discuss how I/O is handled in MPI programs. Explain the issues of input and output with multiple processes.	6	L2	CO3
24.	Explain collective communication in MPI. Describe MPI_Bcast, MPI_Reduce, MPI_Allreduce, MPI_Scatter and MPI_Gather with syntax.	10	L2	CO3
25.	Explain MPI-derived datatypes. Write the code to build a derived datatype for the trapezoidal rule input (a, b, n) and use it in MPI_Bcast.	8	L3	CO3
26.	Explain how the performance of MPI programs is evaluated. Discuss the results of the trapezoidal rule in terms of speedup and efficiency.	8	L2	CO3
27.	Explain a parallel sorting algorithm in MPI (odd-even transposition sort). Discuss its parallel implementation.	10	L3	CO3
28.	Write an MPI program to demonstrate deadlock using point-to-point communication and show how it is avoided by altering the call sequence.	8	L3	CO3
29.	Write an MPI program to demonstrate MPI_Scatter and MPI_Gather with an example of vector addition.	8	L3	CO3
30.	Explain MPI_Reduce and MPI_Allreduce with the operations MPI_MAX, MPI_MIN, MPI_SUM and MPI_PROD. Write an MPI program to demonstrate them.	8	L3	CO3
	Module-04			
31.	Explain OpenMP pragmas and directives. Write an OpenMP 'Hello World' program that uses the parallel directive and discuss the clauses used.	8	L2	CO4
32.	Develop an OpenMP program to implement the trapezoidal rule using the parallel directive and critical section. Explain how the result is combined.	10	L3	CO4
33.	Explain the scope of variables in OpenMP. Discuss the shared, private, default and reduction clauses with examples.	8	L2	CO4

Aditya College of Engineering and Technology

34.	Explain loop carried dependence in OpenMP. Discuss how it can be detected and removed with an example of the Fibonacci sequence or the estimation of pi.	10	L3	CO4
35.	Explain the schedule clause in OpenMP. Differentiate between static, dynamic, guided, auto and runtime scheduling with examples.	8	L2	CO4
36.	Explain the producer-consumer problem in OpenMP with the help of a message-passing (queue) program. Discuss the critical, atomic and lock constructs used.	10	L3	CO4
37.	What is false sharing in OpenMP? Explain caches, cache coherence, tasking and thread safety in OpenMP.	8	L2	CO4
38.	Write an OpenMP program to find the prime numbers from 1 to n using the parallel for directive. Record the serial and parallel execution times.	8	L3	CO4
39.	Write an OpenMP program to calculate n Fibonacci numbers using tasks.	8	L3	CO4
40.	Write an OpenMP program to sort an array of n elements using sequential and parallel merge sort (using sections). Record the difference in execution time.	10	L3	CO4
	Module-05			
41.	Explain GPUs and GPGPU. Discuss the architecture of a GPU and compare it with that of a CPU.	8	L2	CO5
42.	What is heterogeneous computing? Explain the roles of the host and device in a CUDA program.	8	L2	CO5
43.	Explain threads, blocks and grids in CUDA. Discuss the thread hierarchy and how thread indices are computed.	8	L2	CO5
44.	Explain Nvidia compute capabilities and device architectures with respect to streaming multiprocessors and warps.	6	L2	CO5
45.	Write a CUDA program for vector addition. Explain the kernel launch, memory allocation and copying of data between host and device.	10	L3	CO5
46.	Explain how results are returned from CUDA kernels. Develop CUDA trapezoidal rule I and discuss how it is improved in the trapezoidal rule II.	10	L3	CO5
47.	Explain CUDA trapezoidal rule III with blocks having more than one warp. Discuss the performance improvement obtained.	8	L3	CO5
48.	Explain the CUDA programming model: kernels, __global__ functions and the kernel launch configuration <<<blocks, threads>>>.	8	L2	CO5
49.	Explain warps and warp scheduling in an Nvidia GPU. Discuss the effect of branch divergence on performance.	8	L2	CO5
50.	Design a CUDA kernel for the trapezoidal rule that uses shared memory to reduce the partial sums within a block. Explain how it improves performance.	10	L3	CO5

Aditya College of Engineering and Technology

***Note:**

1. All modules should contain a minimum of **10** questions each and the total number of questions must be **50**.
2. Questions can carry marks of 06, 07, 08, 09, 10 and for problems: 04 or 05.
3. Follow the RBTL as per the keywords used to frame the questions.
4. Maintain the Course Outcomes (COs) as per the syllabus.
5. M: Marks, L: RBTL, C:COs.

Aditya College of Engineering and Technology