Name:

PRN No:

**High Performance Computing Lab**

**Practical No. 12**

**Title of practical: Parallel Programming using of CUDA C**

**Problem 1: Vector Addition using CUDA**

Problem Statement: Write a CUDA C program that performs element-wise addition of two vectors A and B of size N. The result of the addition should be stored in vector C.

Details:

* Initialize the vectors A and B with random numbers.
* The output vector C[i] = A[i] + B[i], where i ranges from 0 to N-1.
* Use CUDA kernels to perform the computation in parallel.
* Write the code for both serial (CPU-based) and parallel (CUDA-based) implementations.
* Measure the execution time of both the serial and CUDA implementations for different values of N (e.g., N = 10^5, 10^6, 10^7).

Task:

* Calculate and report the speedup (i.e., the ratio of CPU execution time to GPU execution time).

**Problem 2: Matrix Addition using CUDA**

Problem Statement: Write a CUDA C program to perform element-wise addition of two matrices A and B of size M x N. The result of the addition should be stored in matrix C.

Details:

* Initialize the matrices A and B with random values.
* The output matrix C[i][j] = A[i][j] + B[i][j] where i ranges from 0 to M-1 and j ranges from 0 to N-1.
* Write code for both serial (CPU-based) and parallel (CUDA-based) implementations.
* Measure the execution time of both implementations for various matrix sizes (e.g., 100x100, 500x500, 1000x1000).

Task:

* Calculate the speedup using the execution times of the CPU and GPU implementations.

**Problem 3: Dot Product of Two Vectors using CUDA**

Problem Statement: Write a CUDA C program to compute the dot product of two vectors A and B of size N. The dot product is defined as:

Details:

* Initialize the vectors A and B with random values.
* Implement the dot product calculation using both serial (CPU) and parallel (CUDA) approaches.
* Measure the execution time for both implementations with different vector sizes (e.g., N = 10^5, 10^6, 10^7).
* Use atomic operations or shared memory reduction in the CUDA kernel to compute the final sum.

Task:

* Calculate and report the speedup for different vector sizes.

**Problem 4: Matrix Multiplication using CUDA**

Problem Statement: Write a CUDA C program to perform matrix multiplication. Given two matrices A (MxN) and B (NxP), compute the resulting matrix C (MxP) where:

Details:

* Initialize the matrices A and B with random values.
* Write code for both serial (CPU-based) and parallel (CUDA-based) implementations.
* Measure the execution time of both implementations for various matrix sizes (e.g., 100x100, 500x500, 1000x1000).

Task:

* Calculate the speedup by comparing the CPU and GPU execution times.