**Problem Statement:** Implement Min, Max, Sum and Average operations using Parallel Reduction

**Code:**

#include <iostream>

#include <omp.h>

#include <vector>

#include <climits>

using namespace std;

int main()

{

int n;

cout << "Enter number of elements: ";

cin >> n;

vector<int> arr(n);

cout << "Enter " << n << " elements:\n";

for (int i = 0; i < n; ++i)

cin >> arr[i];

// Sequential Operations

int seq\_min = arr[0], seq\_max = arr[0], seq\_sum = 0;

double seq\_start = omp\_get\_wtime();

for (int i = 0; i < n; ++i)

{

if (arr[i] < seq\_min)

seq\_min = arr[i];

if (arr[i] > seq\_max)

seq\_max = arr[i];

seq\_sum += arr[i];

}

double seq\_end = omp\_get\_wtime();

// Parallel Reduction Operations

int par\_min = INT\_MAX, par\_max = INT\_MIN, par\_sum = 0;

double par\_start = omp\_get\_wtime();

#pragma omp parallel for reduction(min : par\_min) reduction(max : par\_max) reduction(+ : par\_sum)

for (int i = 0; i < n; ++i)

{

par\_min = min(par\_min, arr[i]);

par\_max = max(par\_max, arr[i]);

par\_sum += arr[i];

}

double par\_end = omp\_get\_wtime();

// Output Results

cout << "\nSequential Results:\n";

cout << "Min: " << seq\_min << ", Max: " << seq\_max << ", Sum: " << seq\_sum

<< ", Average: " << (double)seq\_sum / n << endl;

cout << "Time taken: " << (seq\_end - seq\_start) << " seconds\n";

cout << "\nParallel Results (using OpenMP reduction):\n";

cout << "Min: " << par\_min << ", Max: " << par\_max << ", Sum: " << par\_sum

<< ", Average: " << (double)par\_sum / n << endl;

cout << "Time taken: " << (par\_end - par\_start) << " seconds\n";

return 0;

}  
**Steps to run the code:**

1. Run the command “ g++ -fopenmp file\_name.cpp -o file\_name ”
2. Run the command “.\file\_name ”

**Output:**

C:\Users\Shravan\OneDrive\Desktop\Engineering Degree Stuff\4th Year Stuff\8th Sem Stuff\LP-5 Problem Statement & Programs\Programs\HPC Practical 3>g++ -fopenmp parallel\_min\_max.cpp -o parallel\_min\_max

C:\Users\Shravan\OneDrive\Desktop\Engineering Degree Stuff\4th Year Stuff\8th Sem Stuff\LP-5 Problem Statement & Programs\Programs\HPC Practical 3>.\parallel\_min\_max

Enter number of elements: 5

Enter 5 elements:

10 20 5 15 30

Sequential Results:

Min: 5, Max: 30, Sum: 80, Average: 16

Time taken: 0 seconds

Parallel Results (using OpenMP reduction):

Min: 5, Max: 30, Sum: 80, Average: 16

Time taken: 0.00399995 seconds