**Assignment No 3**

Name: Poonam Kisan Salbande

Roll No: 20121011

Title:Implement Min, Max ,Sum and Average operations using Parallel Reduction.Code:-

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <omp.h>

using namespace std;

// Function to generate random array

void generateRandomArray(int arr[], int n) {

srand(time(NULL));

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

arr[i] = rand() % 100;

}

}

// Function to find the minimum value in an array using parallel reduction

int findMin(int arr[], int n) {

int min\_val = arr[0];

#pragma omp parallel for reduction(min:min\_val)

for (int i = 1; i < n; i++) {

if (arr[i] < min\_val) {

min\_val = arr[i];

}

}

return min\_val;

}

// Function to find the maximum value in an array using parallel reduction

int findMax(int arr[], int n) {

int max\_val = arr[0];

#pragma omp parallel for reduction(max:max\_val)

for (int i = 1; i < n; i++) {

if (arr[i] > max\_val) {

max\_val = arr[i];

}

}

return max\_val;

}

// Function to find the sum of values in an array using parallel reduction

int findSum(int arr[], int n) {

int sum = 0;

#pragma omp parallel for reduction(+:sum)

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

sum += arr[i];

}

return sum;

}

// Function to find the average value in an array using parallel reduction

double findAverage(int arr[], int n) {

double avg = 0;

#pragma omp parallel for reduction(+:avg)

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

avg += arr[i];

}

avg /= n;

return avg;

}

int main() {

const int n = 10000;

int arr[n];

// Generate random array

generateRandomArray(arr, n);

// Find minimum value

int min\_val = findMin(arr, n);

cout << "Minimum value: " << min\_val << endl;

// Find maximum value

int max\_val = findMax(arr, n);

cout << "Maximum value: " << max\_val << endl;

// Find sum of values

int sum = findSum(arr, n);

cout << "Sum of values: " << sum << endl;

// Find average value

double avg = findAverage(arr, n);

cout << "Average value: " << avg << endl;

return 0;

}

Output:

