CPP code for Min,Max,Average,Sum

#include <iostream>

//#include <vector> #include <omp.h> #include <climits> using namespace std;

void min\_reduction(int arr[], int n) { int min\_value = INT\_MAX;

#pragma omp parallel for reduction(min: min\_value) for (int i = 0; i < n; i++) {

if (arr[i] < min\_value) { min\_value = arr[i];

}

}

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

}

void max\_reduction(int arr[], int n) { int max\_value = INT\_MIN;

#pragma omp parallel for reduction(max: max\_value) for (int i = 0; i < n; i++) {

if (arr[i] > max\_value) { max\_value = arr[i];

}

}

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

}

void sum\_reduction(int arr[], int n) { int sum = 0;

#pragma omp parallel for reduction(+: sum) for (int i = 0; i < n; i++) {

sum += arr[i];

}

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

}

void average\_reduction(int arr[], int n) { int sum = 0;

#pragma omp parallel for reduction(+: sum) for (int i = 0; i < n; i++) {

sum += arr[i];

}

cout << "Average: " << (double)sum / (n-1) << endl;

}

int main() { int \*arr,n;

cout<<"\n enter total no of elements=>"; cin>>n;

arr=new int[n];

cout<<"\n enter elements=>"; for(int i=0;i<n;i++)

{

cin>>arr[i];

}

// int arr[] = {5, 2, 9, 1, 7, 6, 8, 3, 4};

// int n = size(arr); min\_reduction(arr, n); max\_reduction(arr, n); sum\_reduction(arr, n); average\_reduction(arr, n);

}

Output

