HPC 3 (OpenMP):-

#include <iostream>

#include <vector>

#include <omp.h>

using namespace std;

int parallelMin(vector<int> arr)

{

int min\_val = arr[0];

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

for(int i=1; i<arr.size(); i++)

{

if(arr[i] < min\_val)

min\_val = arr[i];

}

return min\_val;

}

int parallelMax(vector<int> arr)

{

int max\_val = arr[0];

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

for(int i=1; i<arr.size(); i++)

{

if(arr[i] > max\_val)

max\_val = arr[i];

}

return max\_val;

}

int parallelSum(vector<int> arr)

{

int sum = 0;

#pragma omp parallel for reduction(+:sum)

for(int i=0; i<arr.size(); i++)

sum += arr[i];

return sum;

}

double parallelAvg(vector<int> arr)

{

return (double)parallelSum(arr)/arr.size();

}

int main(){

int n;

cout<<"No of elements: ";

cin>>n;

vector<int> arr(n);

cout<<"Enter elements: ";

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

cin>>arr[i];

cout<<"Minimum: "<<parallelMin(arr)<<endl;

cout<<"Maximum: "<<parallelMax(arr)<<endl;

cout<<"Sum: "<<parallelSum(arr)<<endl;

cout<<"Average: "<<parallelAvg(arr)<<endl;

}

OUTPUT:-

~/HPC$ g++ -o hpc3 -fopenmp hpc3.cpp

~/HPC$ ./hpc3

No of elements: 5

Enter elements: 1 2 3 7 8

Minimum: 1

Maximum: 8

Sum: 21

Average: 4.2

~/HPC$