**CODE**

#include <iostream>

#include <omp.h>

#include <ctime>

#include <cstdlib>

#include <climits>  // For INT\_MAX, INT\_MIN

using namespace std;

void min(int \*arr, int n)

{

   int min\_val = INT\_MAX;  // Use INT\_MAX to ensure proper reduction for min

   int i;

   cout << endl;

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

   for (i = 0; i < n; i++)

   {

      // Debugging: showing thread id and iteration index

      // cout << "\nthread id = " << omp\_get\_thread\_num() << " and i = " << i;

      if (arr[i] < min\_val)

      {

         min\_val = arr[i];

      }

   }

   cout << "\n\nmin\_val = " << min\_val << endl;

}

void max(int \*arr, int n)

{

   int max\_val = INT\_MIN;  // Use INT\_MIN to ensure proper reduction for max

   int i;

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

   for (i = 0; i < n; i++)

   {

      // Debugging: showing thread id and iteration index

      // cout << "\nthread id = " << omp\_get\_thread\_num() << " and i = " << i;

      if (arr[i] > max\_val)

      {

         max\_val = arr[i];

      }

   }

   cout << "\n\nmax\_val = " << max\_val << endl;

}

void avg(int \*arr, int n)

{

   int i;

   float avg = 0.0, sum = 0;

   #pragma omp parallel for reduction(+:sum)

   for (i = 0; i < n; i++)

   {

      sum += arr[i];

      // Debugging: showing thread id and iteration index

      // cout << "\nthread id = " << omp\_get\_thread\_num() << " and i = " << i;

   }

   cout << "\n\nSum = " << sum << endl;

   avg = sum / n;

   cout << "\nAverage = " << avg << endl;

}

int main()

{

   omp\_set\_num\_threads(4);  // Set the number of threads

   int n, i;

   cout << "Enter the number of elements in the array: ";

   cin >> n;

   int arr[n];  // Create an array of size n

   srand(time(0));  // Initialize random seed

   for (i = 0; i < n; ++i)

   {

      arr[i] = rand() % 100;  // Random values between 0 and 99

   }

   cout << "\nArray elements are: ";

   for (i = 0; i < n; i++)

   {

      cout << arr[i] << " ";

   }

   cout << endl;

   min(arr, n);

   max(arr, n);

   avg(arr, n);

   return 0;

}

**OUTPUT**

D:\Study Material\sem8\ASSIGNMENTS\HPC\3>g++ -fopenmp 3.cpp

D:\Study Material\sem8\ASSIGNMENTS\HPC\3>a.exe

Enter the number of elements in the array: 20

Array elements are: 95 71 50 6 65 8 3 75 57 89 58 14 49 49 53 50 39 12 34 97

min\_val = 3

max\_val = 97

Sum = 974

Average = 48.7