**Assignment 3**

**Min, Max, Sum & Average operations using Parallel Reduction**

#include<stdio.h>

#include<omp.h>

int main()

{

omp\_set\_num\_threads(4);

double arr[10]={1,2,3,4,5,6,7,8,9,10};

double max\_val=0.0;

double min\_val=100.0;

float avg=0.0,sum=0.0,sum\_val=0.0;

int i;

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

for(i=0;i<10;i++)

{

printf("thread id = %d and i = %d \n", omp\_get\_thread\_num(),i);

if(arr[i] < min\_val)

{

min\_val = arr[i];

}

}

printf("min\_val = %f", min\_val);

printf("\n");

printf("\n");

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

for(i=0;i<10;i++)

{

printf("thread id = %d and i = %d \n", omp\_get\_thread\_num(),i);

if(arr[i] > max\_val)

{

max\_val = arr[i];

}

}

printf("max\_val = %f", max\_val);

printf("\n");

printf("\n");

#pragma omp parallel for reduction(+:sum\_val)

for(i=0;i<10;i++)

{

printf("thread id = %d and i = %d \n", omp\_get\_thread\_num(),i);

sum\_val=sum\_val+arr[i];

}

printf("sum\_val = %f", sum\_val);

printf("\n");

printf("\n");

#pragma omp parallel for reduction(+:sum)

for(i=0;i<10;i++)

{

printf("thread id = %d and i = %d \n", omp\_get\_thread\_num(),i);

sum=sum+arr[i];

}

avg=sum/10;

printf("avg\_val = %f", avg);

printf("\n");

printf("\n");

}

**Output –**

thread id = 3 and i = 8

thread id = 3 and i = 9

thread id = 1 and i = 3

thread id = 1 and i = 4

thread id = 1 and i = 5

thread id = 2 and i = 6

thread id = 2 and i = 7

thread id = 0 and i = 0

thread id = 0 and i = 1

thread id = 0 and i = 2

min\_val = 1.000000

thread id = 1 and i = 3

thread id = 1 and i = 4

thread id = 1 and i = 5

thread id = 3 and i = 8

thread id = 3 and i = 9

thread id = 0 and i = 0

thread id = 0 and i = 1

thread id = 0 and i = 2

thread id = 2 and i = 6

thread id = 2 and i = 7

max\_val = 10.000000

thread id = 3 and i = 8

thread id = 3 and i = 9

thread id = 2 and i = 6

thread id = 2 and i = 7

thread id = 1 and i = 3

thread id = 1 and i = 4

thread id = 1 and i = 5

thread id = 0 and i = 0

thread id = 0 and i = 1

thread id = 0 and i = 2

sum\_val = 55.000000

thread id = 1 and i = 3

thread id = 1 and i = 4

thread id = 1 and i = 5

thread id = 0 and i = 0

thread id = 0 and i = 1

thread id = 0 and i = 2

thread id = 3 and i = 8

thread id = 3 and i = 9

thread id = 2 and i = 6

thread id = 2 and i = 7

avg\_val = 5.500000