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