NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA)

LEARNING ASSESSMENT 1 REPORT

High Performance Computing

(18CS71)

Submitted in partial fulfilment of the requirement for the award of Degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

V Venkata Sree Harsha	1NT18CS181

NITTE EDUCATION TRUST

Department of Computer Science and Engineering

2021-22

TABLE OF CONTENTS

PROBLEM STATEMENT

A school has five sections for each grade and they have three grades (PreKG, LKG and UKG). They send a teacher to each section to get the information about the age of each child in the section. Assume that there are ten students in each section. Assuming each teacher is simulated as a thread use the MPI-Reduction function to determine the Minimum age, the Maximum age and the Average age for each section, each grade and for the whole school. Use random numbers for generating the age of the children.

3 standrards - PreKG, LKG, UKG

5 sections -

PreKG - A B C D E

LKG - A B C D E

UKG - A B C D E

Total = 5*3 = 15

Each section has 10 students

so total students 10*15 = 150 students

Each teacher is assigned 1 section to find local_min and local_max and local_avg of the ages

All teachers should combine the data and the communicator will find the global_min and global_max and global_avg

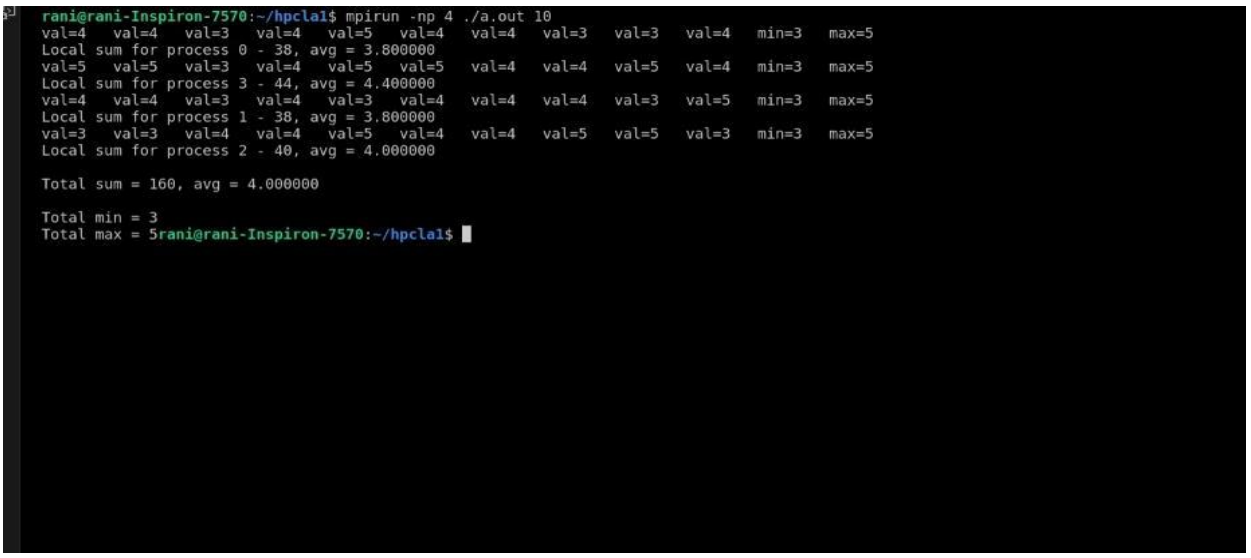
So we need to do MPI_reduce

Solution Code

```
#include <stdio.h>
#include <stdlib.h>
#include <mpi.h>
#include <assert.h>
#include <time.h>
int min_arr(int arr[], int n)
{
    int i;
    int min = arr[0];
    for (i = 1; i < n; i++)
        if (arr[i] < min)
            min = arr[i];

    return min;
}
int largest(int arr[], int n)
{
    int i;
    int max = arr[0];

    for (i = 1; i < n; i++)
        if (arr[i] > max)
            max = arr[i];

    return max;
}
int *create_rand_nums(int num_elements) {
  int *rand_nums = (int *)malloc(sizeof(int) * num_elements);
  assert(rand_nums != NULL);
  int i;
  for (i = 0; i < num_elements; i++) {
    rand_nums[i] = (rand() % (5 - 3 + 1) + 3);
    printf("val=%d\t",rand_nums[i]);
  }

  return rand_nums;
}

int main(int argc, char** argv) {
  if (argc != 2) {
    fprintf(stderr, "Usage: avg num_elements_per_proc\n");
    exit(1);
  }

  int num_elements_per_proc = atoi(argv[1]);

  MPI_Init(NULL, NULL);

  int world_rank;
  MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
  int world_size;
  MPI_Comm_size(MPI_COMM_WORLD, &world_size);

  srand(time(NULL)*world_rank);
  int *rand_nums = NULL;
  rand_nums = create_rand_nums(num_elements_per_proc);
```

```
 int local_min=min_arr(rand_nums,num_elements_per_proc);
 printf("min=%d\t",local_min);

 int local_max=largest(rand_nums,num_elements_per_proc);
 printf("max=%d\t",local_max);

 int local_sum = 0;
 int i;
 for (i = 0; i < num_elements_per_proc; i++) {
   local_sum += rand_nums[i];
 }

 float avg=(float)local_sum / (float)num_elements_per_proc;
 printf("\nLocal sum for process %d - %d, avg = %f\n",world_rank,local_sum,avg);

 int global_sum;
 MPI_Reduce(&local_sum, &global_sum, 1, MPI_INT, MPI_SUM, 0,
           MPI_COMM_WORLD);
 int global_min;
 MPI_Reduce(&local_min, &global_min, 1, MPI_INT, MPI_MIN, 0,
           MPI_COMM_WORLD);
int global_max;
 MPI_Reduce(&local_max, &global_max, 1, MPI_INT, MPI_MAX, 0,
           MPI_COMM_WORLD);

 if (world_rank == 0) {
  float avg1= (float)global_sum / (float)(world_size * num_elements_per_proc);
   printf("\nTotal sum = %d, avg = %f\n", global_sum,avg1);
   printf("\nTotal min = %d", global_min);
printf("\nTotal max = %d", global_max);
 }
 free(rand_nums);

 MPI_Barrier(MPI_COMM_WORLD);
 MPI_Finalize();

 //CODE WRITTEN BY VALLAMKONDA VENKATA SREE HARSHA
 //USN: 1NT18CS181
 //SECTION: B

}
```

Output:


rani@rani-Inspiron-7570:~/hpcla1$ mpirun -np 4 ./a.out 10
val=4 val=4 val=3 val=4 val=5 val=4 val=4 val=3 val=3 val=4 min=3 max=5
Local sum for process 0 - 38, avg = 3.800000
val=5 val=5 val=3 val=4 val=5 val=5 val=4 val=4 val=5 val=4 min=3 max=5
Local sum for process 3 - 44, avg = 4.400000
val=4 val=4 val=3 val=4 val=3 val=4 val=4 val=4 val=3 val=5 min=3 max=5
Local sum for process 1 - 38, avg = 3.800000
val=3 val=3 val=4 val=4 val=5 val=4 val=4 val=5 val=5 val=3 min=3 max=5
Local sum for process 2 - 40, avg = 4.000000
Total sum = 160, avg = 4.000000
Total min = 3
Total max = 5rani@rani-Inspiron-7570:~/hpcla1$