

Question 27 - Mean and Variance using Parallel Computing

Mean:

```
#include "mpi.h"
#include <stdio.h>
#include <math.h>
#include <cstdlib>
#include <iostream>

int main( int argc, char *argv[] )
{
    int myid, numprocs;
    int count, mycount = 0;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);
    int m=10000;

    double sum = 0;
    double sumAll = 0;
    for(int i = 0; i<m; i++)
    {
        double x = (double)rand()/(RAND_MAX + 1.0);
        sum += x;
    };

    MPI_Reduce(&sum, &sumAll, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);

    if(myid==0)
    {
        double meanvalue = sumAll/((double)(numprocs*m));
        std::cout<<"meanvalue = "<<meanvalue<<"\n";
        std::cout<<"number of simulations = "<<numprocs*m;
    }

    MPI_Finalize();
    return 0;
}
```

Variance:

```

#include "mpi.h"
#include <stdio.h>
#include <math.h>
#include <cstdlib>
#include <iostream>

int main( int argc, char *argv[] )
{
    int myid, numprocs;
        int count,mycount = 0;
    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD,&myid);
        int m=10000;

        double sum = 0;
        double sumAll = 0;
        double sum2 = 0;
        double sum2All = 0;
        for(int i = 0;i<m;i++)
        {
            double x = (double)rand()/(RAND_MAX + 1.0);
            sum += x;
            sum2 += x*x;
        };

        MPI_Reduce(&sum, &sumAll, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);
        MPI_Reduce(&sum2, &sum2All, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);

        if(myid==0)
        {
            int n = numprocs*m;
            double meanvalue = sumAll/((double)(n));
            double variance = (sum2All - sumAll*sumAll/n)/(n
- 1.0);

            std::cout<<"meanvalue = "<<meanvalue<<"\n";
            std::cout<<"variance = "<<variance<<"\n";
            std::cout<<"sumAll = "<<sumAll<<"\n";
            std::cout<<"sum2All = "<<sum2All<<"\n";
            std::cout<<"number of simulations = "<<n;
        }

    MPI_Finalize();
    return 0;
}

```

