VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Computer Science & Engineering

CSE4001

Parallel and Distributed Computing

LAB ASSIGNMENT 9

Submitted to **Prof. DEEBAK B.D.**

TOPIC: PROBLEMS USING MPI

NAME: PUNIT MIDDHA

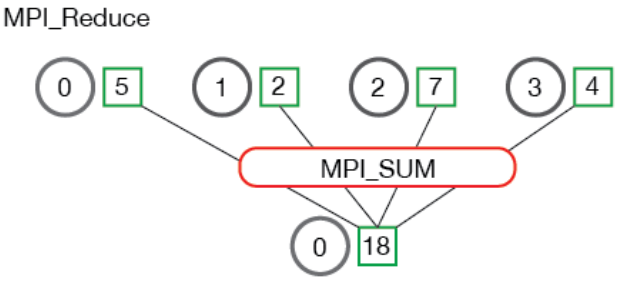
REG.NO: 19BCE2060

SLOT: L55+L56

DATE: 30/11/2021

QUESTION – I

Write a C program to use MPI_Reduce that divides the processors into the group to find the addition independently.



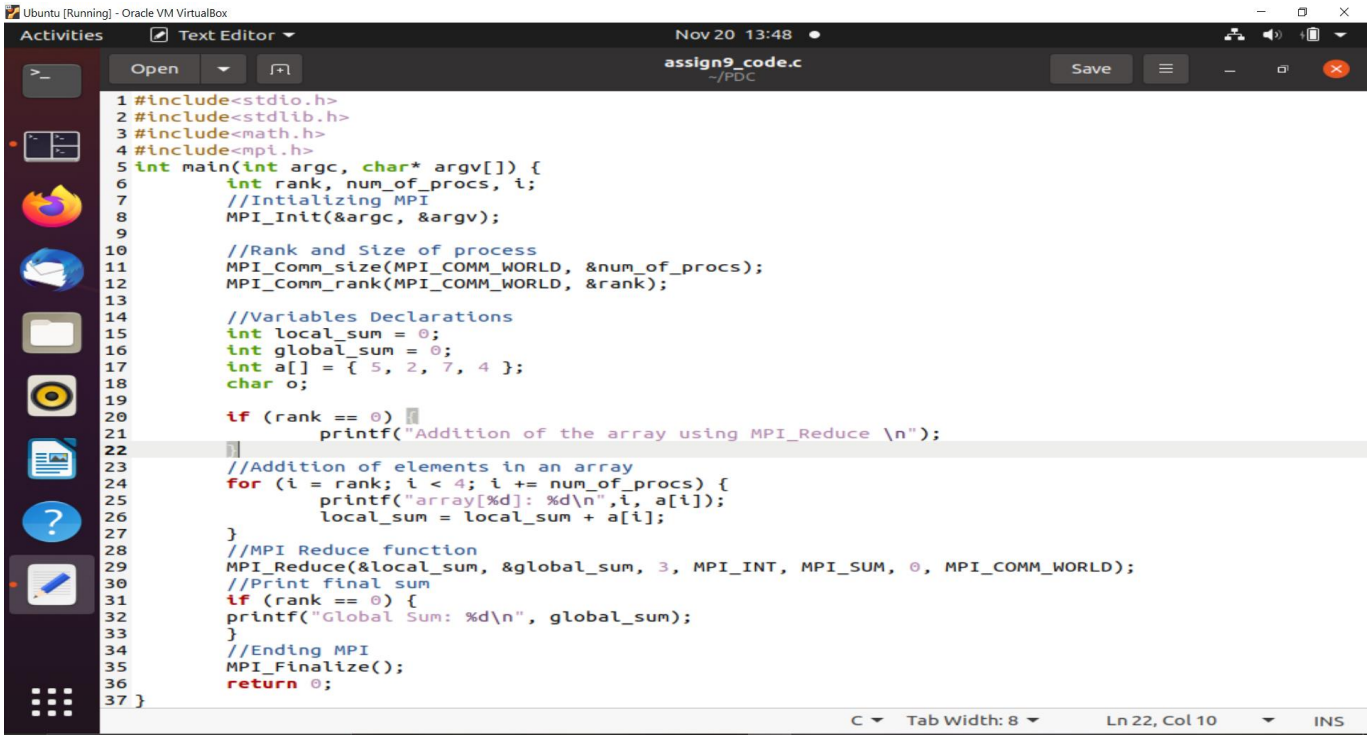
Hint. The function prototype is as follows:

```
MPI_Reduce(
void* send_data,
void* recv_data,              int
count,
                MPI_Datatype datatype,
              MPI_Op op,
int root,
                MPI_Comm communicator)
```

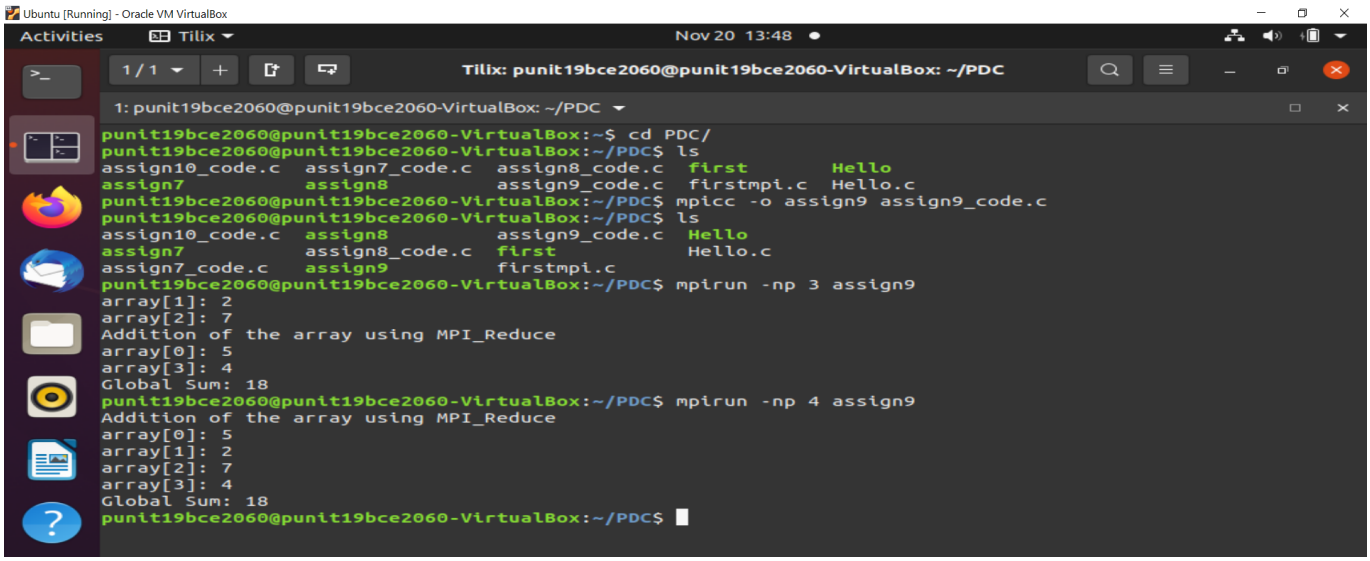
SOURCE CODE:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<mpi.h>
int main(int argc, char* argv[]) {
int rank, num_of_procs, i;
//Intializing MPI
MPI_Init(&argc, &argv);

//Rank and Size of process
MPI_Comm_size(MPI_COMM_WORLD, &num_of_procs);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);

//Variables Declarations
int local_sum = 0;
int global_sum = 0;
int a[] = { 5, 2, 7, 4 };
char o;

if (rank == 0) {
printf("Addition of the array using MPI_Reduce \n");
}
//Addition of elements in an array
```

```
for (i = rank; i < 4; i += num_of_procs) {
printf("array[%d]: %d\n",i, a[i]);
local_sum = local_sum + a[i];
}
//MPI Reduce function
MPI_Reduce(&local_sum, &global_sum, 3, MPI_INT, MPI_SUM, 0, MPI_COMM_WORLD);
//Print final sum
if (rank == 0) {
printf("Global Sum: %d\n", global_sum);
}
//Ending MPI
MPI_Finalize();
return 0;
}
```



EXECUTION:



REMARKS:

- ✓ The purpose of this experiment was to learn about the MPI Reduce() function.
- ✓ The reduce function speeds up the calculation for all processes in a collection.
- ✓ The reduction operation is used to compute the items sent by the MPI, and the result is saved in the root.
- ✓ In the above implementation, the root process computes the sum of each process's array items, which is subsequently shown in the Global sum variable.