PARALLEL AND DISTRIBUTED COMPUTING LAB

REPORT

NAME: S Shyam Sundaram

REG NO: 19BCE1560

PROGRAMMING ENVIRONMENT: MPI

PROBLEM: MPI

DATE: 20th October, 2021

HARDWARE CONFIGURATION:

CPU NAME	:	Intel core i5 – 1035G1 @ 1.00 Ghz
Number of Sockets:	:	1
Cores per Socket	:	4
Threads per core	:	1
L1 Cache size	:	320KB
L2 Cache size	:	2MB
L3 Cache size (Shared):		6MB
RAM	:	8 GB

QUESTION 1

Write an MPI program that performs matrix multiplication.

CODE

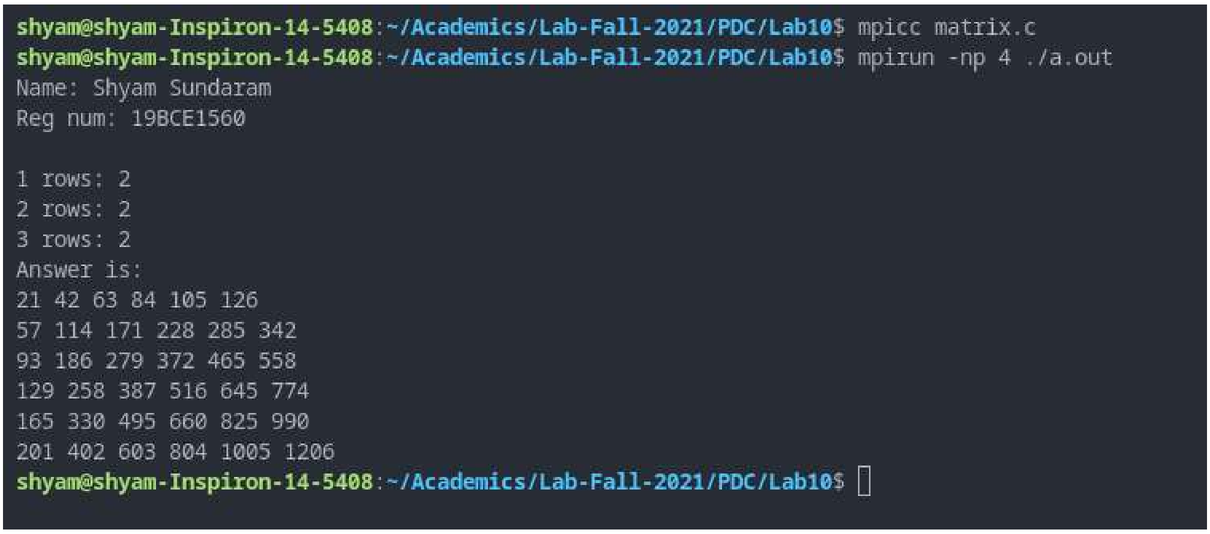
```
#include <mpi.h>
#include <stdio.h>

#define N 12 //can change this

int a[N][N],b[N][N],c[N][N];

MPI_Status status;

int main(int argc, char**argv)
{
   MPI_Init(&argc,&argv);
   int id,numprocs;

   MPI_Comm_rank(MPI_COMM_WORLD,&id);
   MPI_Comm_size(MPI_COMM_WORLD,&numprocs);

   if(id==0) //master
```

```
{
    printf("Name: Shyam Sundaram\nReg num: 19BCE1560\n\n");
    int co=1;
    for(int i=0;i<N;++i)
    for(int j=0;j<N;++j)
    {
        a[i][j]=co++;
        b[i][j]=j+1;
    }

    int rows=N/(numprocs-1),off=0;

    for(int i=1;i<numprocs;++i)
    {
        MPI_Send(&off,1,MPI_INT,i,1,MPI_COMM_WORLD);
        MPI_Send(&rows,1,MPI_INT,i,1,MPI_COMM_WORLD);
        MPI_Send(&a[off],rows*N,MPI_INT,i,1,MPI_COMM_WORLD);
        MPI_Send(&b,N*N,MPI_INT,i,1,MPI_COMM_WORLD);
        off=off+rows;
    }

    for(int i=1;i<numprocs;++i)
    {
        MPI_Recv(&off,1,MPI_INT,i,2,MPI_COMM_WORLD,&status);
        MPI_Recv(&rows, 1, MPI_INT, i, 2, MPI_COMM_WORLD, &status);
        MPI_Recv(&c[off], rows*N, MPI_DOUBLE, i, 2, MPI_COMM_WORLD, &status);
    }
    printf("Answer is: \n");

    for (int i=0; i<N; i++)
    {
        for (int j=0; j<N; j++)
            printf("%d ", c[i][j]);
        printf ("\n");
    }
}

else //workers
{
    int off,rows;
    MPI_Recv(&off, 1, MPI_INT, 0, 1, MPI_COMM_WORLD, &status);
    MPI_Recv(&rows, 1, MPI_INT, 0, 1, MPI_COMM_WORLD, &status);
    MPI_Recv(&a, rows*N, MPI_DOUBLE, 0, 1, MPI_COMM_WORLD, &status);
    MPI_Recv(&b, N*N, MPI_DOUBLE, 0, 1, MPI_COMM_WORLD, &status);
    printf("%d rows: %d\n",id,rows);
    /* Matrix multiplication */
    for (int k=0; k<N; k++)
    for (int i=0; i<rows; i++)
    {
      c[i][k] = 0;
      for (int j=0; j<N; j++)
        c[i][k] = c[i][k] + a[i][j] * b[j][k];
    }

    MPI_Send(&off, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
    MPI_Send(&rows, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
    MPI_Send(&c, rows*N, MPI_DOUBLE, 0, 2, MPI_COMM_WORLD);
  }

  MPI_Finalize();
}
```

COMMANDS

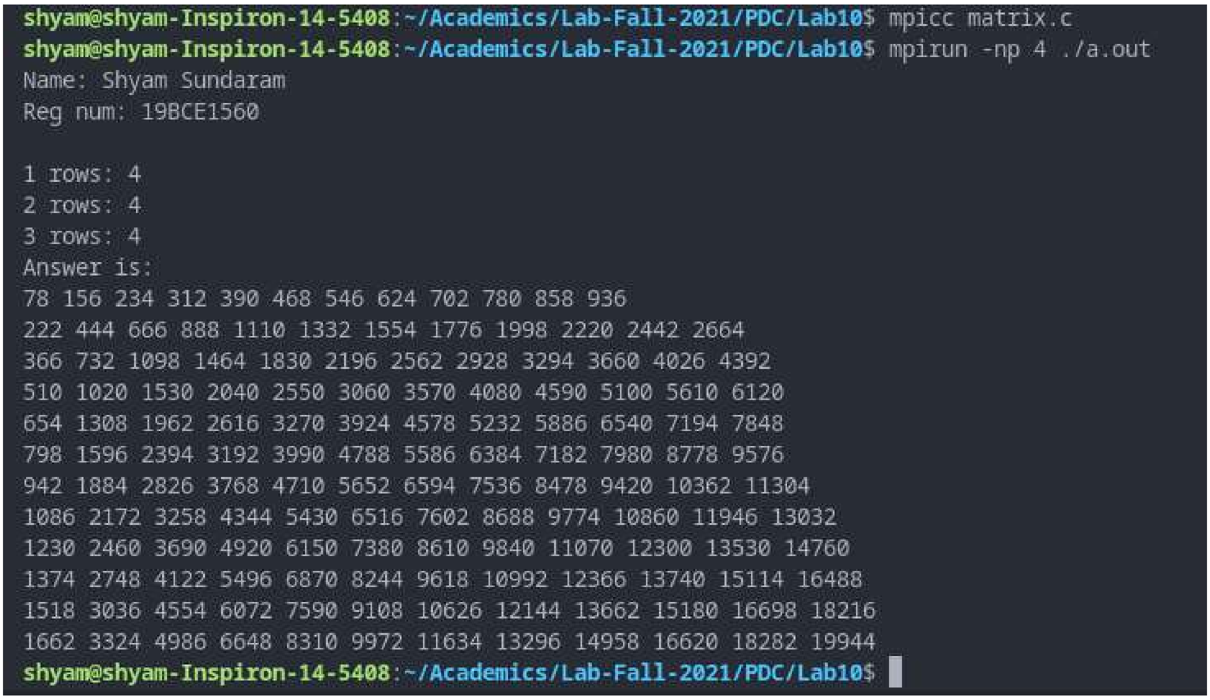
```
mpicc matrix.c
mpirun -np 4 ./a.out
```

OUTPUT



```
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$ mpicc matrix.c
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$ mpirun -np 4 ./a.out
Name: Shyam Sundaram
Reg num: 19BCE1560

1 rows: 2
2 rows: 2
3 rows: 2
Answer is:
21 42 63 84 105 126
57 114 171 228 285 342
93 186 279 372 465 558
129 258 387 516 645 774
165 330 495 660 825 990
201 402 603 804 1005 1206
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$ 
```

With N=6



```
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$ mpicc matrix.c
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$ mpirun -np 4 ./a.out
Name: Shyam Sundaram
Reg num: 19BCE1560

1 rows: 4
2 rows: 4
3 rows: 4
Answer is:
78 156 234 312 390 468 546 624 702 780 858 936
222 444 666 888 1110 1332 1554 1776 1998 2220 2442 2664
366 732 1098 1464 1830 2196 2562 2928 3294 3660 4026 4392
510 1020 1530 2040 2550 3060 3570 4080 4590 5100 5610 6120
654 1308 1962 2616 3270 3924 4578 5232 5886 6540 7194 7848
798 1596 2394 3192 3990 4788 5586 6384 7182 7980 8778 9576
942 1884 2826 3768 4710 5652 6594 7536 8478 9420 10362 11304
1086 2172 3258 4344 5430 6516 7602 8688 9774 10860 11946 13032
1230 2460 3690 4920 6150 7380 8610 9840 11070 12300 13530 14760
1374 2748 4122 5496 6870 8244 9618 10992 12366 13740 15114 16488
1518 3036 4554 6072 7590 9108 10626 12144 13662 15180 16698 18216
1662 3324 4986 6648 8310 9972 11634 13296 14958 16620 18282 19944
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$
```

With N=12

OBSERVATION

Each worker worked on multiplying the respective rows they received from matrix ‘a’ and multiplied with matrix ‘b’ and returned the corresponding resultant row to the master.

CONCLUSION

We have found product of matrices using MPI in C.

QUESTION 2

Write an MPI program that counts the number of primes in a given range.

CODE

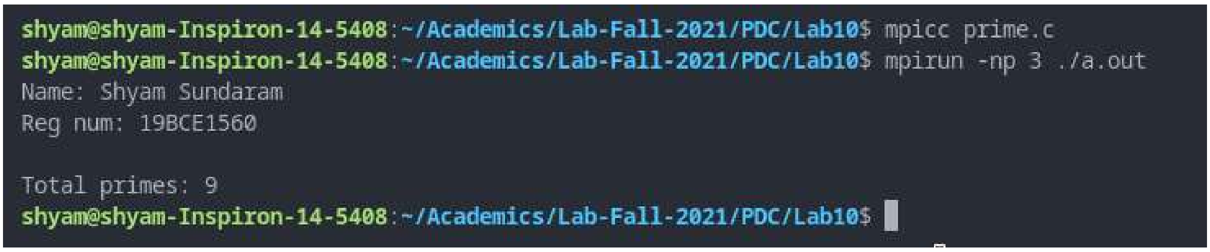
```
#include <mpi.h>
#include <stdio.h>

MPI_Status status;

int isPrime(int n)
{
   for(int i=2;i*i<=n;++i)
   {
      if(n%i==0)
      return -1;
   }
   return 1;
}

int main(int argc, char** argv)
{
   MPI_Init(&argc,&argv);

   int id,numprocs;
   MPI_Comm_rank(MPI_COMM_WORLD,&id);
   MPI_Comm_size(MPI_COMM_WORLD,&numprocs);

   if(id==0)   //master
   {
      int s=1,e=1000; //s is the start of range and e is the end. They are inclusive.
      int num=e-s+1;
      int pernum=num/(numprocs-1);
      int off=s;
      int count=0,ret;
      int leftover=num%(numprocs-1);

      for(int i=1;i<numprocs;++i)
      {
         MPI_Send(&pernum,1,MPI_INT,i,1,MPI_COMM_WORLD);
         MPI_Send(&off,1,MPI_INT,i,1,MPI_COMM_WORLD);
         off=off+pernum;
      }

      for(int i=1;i<numprocs;++i)
```

```
    {
      MPI_Recv(&ret,1,MPI_INT,i,2,MPI_COMM_WORLD,&status);
      count+=ret;
    }

    if(leftover>0)
    {
      for(int i=off;i<off+leftover;++i)
      {
        if(i!=1 && isPrime(i)==1)
        count++;
      }
    }

    printf("Name: Shyam Sundaram\nReg num: 19BCE1560\n\n");
    printf("Total primes: %d\n",count);
  }
  else
  {
    int n,s,count=0;
    MPI_Recv(&n,1,MPI_INT,0,1,MPI_COMM_WORLD,&status);
    MPI_Recv(&s,1,MPI_INT,0,1,MPI_COMM_WORLD,&status);
    for(int i=s;i<s+n;++i)
    {
      //printf("ID %d: %d %d\n ",id,i,isPrime(i));
      if(i!=1 && isPrime(i)==1)
      count++;
    }
    MPI_Send(&count, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
  }

  MPI_Finalize();

}
```

COMMANDS

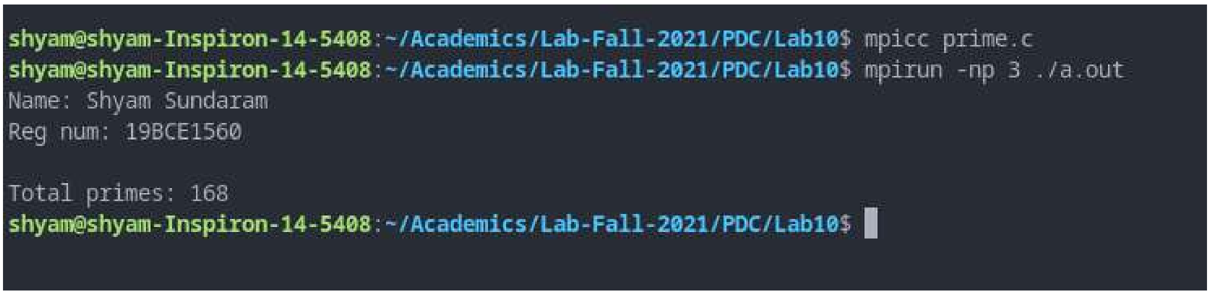
```
mpicc prime.c
mpirun -np 4 ./a.out
```

OUTPUT



```
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$ mpicc prime.c
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$ mpirun -np 3 ./a.out
Name: Shyam Sundaram
Reg num: 19BCE1560

Total primes: 9
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$
```

Between 2 and 23



```
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$ mpicc prime.c
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$ mpirun -np 3 ./a.out
Name: Shyam Sundaram
Reg num: 19BCE1560

Total primes: 168
shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab10$
```

Between 1 and 1000

OBSERVATION

Each worker worked in a specific sub-range given to them and returned number of primes to the master. The master in turns sums the counts and also counts the left-over primes.

CONCLUSION

We have found total number of primes in a given range using MPI in C.