

PARALLEL PROGRAMMING LAB-1 SUBMISSION

NAME : DEVADATHAN N R

ROLL NO : 04

SECTION : CSE A1

REG NO : 230905010

DATE : 09/01/2026

WEEK 1

Question 0 (Sample Question):

```
/*
```

```
Program : Hello World
```

```
Name : Devadathan N R
```

```
Class : CSE - A
```

```
Roll No : 04
```

```
Reg No : 230905010
```

```
*/
```

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

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

```
int main(int argc,char *argv[]){  
    int rank,size;
```

```
    MPI_Init(&argc,&argv);
```

```
    MPI_Comm_size(MPI_COMM_WORLD, &size);
```

```
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
```

```
    printf("Processor %d of %d: Hello World!\n",rank,size);
```

```
    MPI_Finalize();
```

```
    return 0;
```

```
}
```

Output:

```
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpicc l1s1.c -o l1s1
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpirun -n 4 ./l1s1
Processor 0 of 4: Hello World!
Processor 2 of 4: Hello World!
Processor 1 of 4: Hello World!
Processor 3 of 4: Hello World!
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$
```

Question 1 Part 1:

```
/*  
Program : Hello World based on rank  
Name : Devadathan N R  
Class : CSE - A  
Roll No : 04  
Reg No : 230905010  
*/  
  
#include<stdio.h>  
#include<mpi.h>  
  
int main(int argc,char *argv[]){  
    int rank;  
  
    MPI_Init(&argc,&argv);  
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);  
  
    if(rank%2==0)printf("Hello\n");  
    else printf("World\n");  
  
    MPI_Finalize();  
    return 0;  
}
```

Output:

```
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpicc l1q1p1.c -o l1q1p1  
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpirun -n 4 ./l1q1p1  
Hello  
World  
Hello  
World  
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$
```

Question 1 Part 2:

```
/*
Program : pow(x,rank)
Name : Devadathan N R
Class : CSE - A
Roll No : 04
Reg No : 230905010
*/

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

int main(int argc, char *argv[]) {
    int rank, x=5;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    printf("Rank %d output -> pow(%d, %d): %f\n", rank, x, rank, pow(x,
rank));

    MPI_Finalize();
    return 0;
}
```

Output:

```
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpicc l1q1p2.c -o l1q1p2 -lm
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpirun -n 4 ./l1q1p2
Rank 1 output -> pow(5, 1): 5.000000
Rank 3 output -> pow(5, 3): 125.000000
Rank 0 output -> pow(5, 0): 1.000000
Rank 2 output -> pow(5, 2): 25.000000
```

Question 2:

```
/*
Program : Simple Calculator Simulation
Name : Devadathan N R
Class : CSE - A
Roll No : 04
Reg No : 230905010
*/

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

int main(int argc, char *argv[])
{
    int rank, size, a, b;
    a=10;
    b=5;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if (rank == 0) {
        printf("Addition: %d + %d = %d\n", a, b, a + b);
    }
    else if (rank == 1) {
        printf("Subtraction: %d - %d = %d\n", a, b, a - b);
    }
    else if (rank == 2) {
        printf("Multiplication: %d * %d = %d\n", a, b, a * b);
    }
    else if (rank == 3) {
        printf("Division: %d / %d = %d\n", a, b, a / b);
    }

    MPI_Finalize();
    return 0;
}
```

Output:

```
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpicc l1q2.c -o l1q2
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpirun -n 4 ./l1q2
Addition: 10 + 5 = 15
Subtraction: 10 - 5 = 5
Multiplication: 10 * 5 = 50
Division: 10 / 5 = 2
```

Question 3:

```
/*
Program : Rank based string charecter toggle
Name : Devadathan N R
Class : CSE - A
Roll No : 04
Reg No : 230905010
*/

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

int main(int argc, char *argv[])
{
    int rank, size;
    char str[] = "HELLO";
    int len;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    len = strlen(str);

    if (rank < len) {
        if (str[rank] >= 'A' && str[rank] <= 'Z')
            str[rank] = str[rank] + 32;
        else if (str[rank] >= 'a' && str[rank] <= 'z')
            str[rank] = str[rank] - 32;

        printf("Process %d toggled string: %s\n", rank, str);
    }

    MPI_Finalize();
    return 0;
}
```

Output:

```
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpicc l1q3.c -o l1q3
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpirun -n 4 ./l1q3
Process 0 toggled string: hELLO
Process 2 toggled string: HEllO
Process 1 toggled string: HeLLo
Process 3 toggled string: HELlO
```

Question 4:

```
/*
Program : Even Factorial Odd Fibonacci
Name : Devadathan N R
Class : CSE - A
Roll No : 04
Reg No : 230905010
*/

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

long factorial(int n){
    long f = 1;
    for (int i = 1; i <= n; i++)
        f *= i;
    return f;
}

long fibonacci(int n) {
    if (n == 0) return 0;
    if (n == 1) return 1;

    long a = 0, b = 1, c;
    for (int i = 2; i <= n; i++) {
        c = a + b;
        a = b;
        b = c;
    }
    return b;
}

int main(int argc, char *argv[]) {
    int rank, size;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if (rank % 2 == 0) {
        printf("factorial(%d) = %ld\n", rank, factorial(rank));
    } else {
        printf("fibonacci(%d) = %ld\n", rank, fibonacci(rank));
    }

    MPI_Finalize();
    return 0;
}
```

Output:

```
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpicc l1q4.c -o l1q4
STUDENT@MIT-ICT-LAB5-4:~/Desktop/230905010/WEEK1$ mpirun -n 4 ./l1q4
factorial(0) = 1
fibonacci(1) = 1
factorial(2) = 2
fibonacci(3) = 2
```