

Q1

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

int power(int base, int exponent)
{
    if (base == 0 || base == 1)
        return base;
    else if (exponent == 0)
        return 1;
    else if (exponent == 1)
        return base;
    else
        return base * power(base, exponent - 1);
}

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);

    int x = 10;
    printf("x = %d => pow(x,%d) is %d \n", x, rank, power(x, rank));

    MPI_Finalize();
    return 0;
}
```

```
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$ mpicc lab01_q1.c -o lab01_q1 && mpirun -np 4 ./lab01_q1
x = 10 => pow(x,0) is 1
x = 10 => pow(x,1) is 10
x = 10 => pow(x,2) is 100
x = 10 => pow(x,3) is 1000
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$
```

Ln 13, Col 26 Spaces: 4 U

Q2

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

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("Hello (Rank %d) \n", rank);
    else
        printf("World (Rank %d) \n", rank);

    MPI_Finalize();
    return 0;
}
```

```
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$ mpicc lab01_q2.c -o lab01_q2 && mpirun -np 4 ./lab01_q2
Hello (Rank 0)
World (Rank 1)
Hello (Rank 2)
World (Rank 3)
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$
```

Ln 13, Col 26 Spaces: 4

Q3

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

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

    const int num1 = 10;
    const int num2 = 20;

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

    int result = 0;
    switch (rank)
    {
    case 0:
        result = num1 + num2;
        printf("%d + %d is %d (Rank %d) \n", num1, num2, (num1 + num2), rank);
        break;
    case 1:
        result = num1 - num2;
        printf("%d - %d is %d (Rank %d) \n", num1, num2, (num1 - num2), rank);
        break;
    case 2:
        result = num1 * num2;
        printf("%d * %d is %d (Rank %d) \n", num1, num2, (num1 * num2), rank);
        break;
    case 3:
        result = num1 / num2;
        printf("%d / %d is %d (Rank %d) \n", num1, num2, (num1 / num2), rank);
        break;
    }

    MPI_Finalize();
    return 0;
}
```

```
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$ mpicc lab01_q3.c -o lab01_q3 && mpirun -np 4 ./lab01_q3
10 + 20 is 30 (Rank 0)
10 - 20 is -10 (Rank 1)
10 * 20 is 200 (Rank 2)
10 / 20 is 0 (Rank 3)
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$
```

Lab 12 - Col 16 - Spaces 4 - Un

Q4

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

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

    char str[] = "HeLL0";

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

    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("My rank is %d and the modified string is %s \n", rank, str);

    MPI_Finalize();
    return 0;
}
```

```
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$ mpicc lab01_q4.c -o lab01_q4 && mpirun -np 5 ./lab01_q4
My rank is 0 and the modified string is heLL0
My rank is 2 and the modified string is HeLL0
My rank is 3 and the modified string is HeLL0
My rank is 1 and the modified string is HELLO
My rank is 4 and the modified string is HeLL0
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$
```

Ln 13, Col 26 Spaces: 4 UTF

Q1 Additional Exercise

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

int reverse_digits(int num)
{
    int rev = 0;

    while (num > 0)
    {
        int last = num % 10;
        rev = rev * 10 + last;
        num = num / 10;
    }

    return rev;
}

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);

    int input[] = {18, 523, 301, 1234, 2, 14, 108, 150, 1928};

    switch (rank)
    {
        case 0:
        case 1:
        case 2:
        case 3:
        case 4:
        case 5:
        case 6:
        case 7:
        case 8:
            printf("after reversing %d , we get %d (Rank %d) \n", input[rank],
reverse_digits(input[rank]), rank);
            break;

        default:
            break;
    }

    MPI_Finalize();
    return 0;
}
```

```
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$ mpicc lab01_additional.c -o lab01_additional && mpirun -np 9 ./lab01_additi
onall
after reversing 18 , we get 81 (Rank 0)
after reversing 301 , we get 103 (Rank 2)
after reversing 2 , we get 2 (Rank 4)
after reversing 14 , we get 41 (Rank 5)
after reversing 1928 , we get 8291 (Rank 8)
after reversing 108 , we get 801 (Rank 6)
after reversing 150 , we get 51 (Rank 7)
after reversing 1234 , we get 4321 (Rank 3)
after reversing 523 , we get 325 (Rank 1)
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$
```

Q2 Additional Exercise

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

bool isPrime(int num)
{
    if (num < 2)
        return false;

    for (int i = 2; i < num; i++)
    {
        if (num % i == 0)
            return false;
    }

    return true;
}

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 == 0)
    {
        for (int i = 1; i < 50; i++)
        {
            if (isPrime(i))
                printf("Prime number %d (Rank %d ) \n", i, rank);
        }
    }
    else
    {
        for (int i = 51; i < 101; i++)
        {
            if (isPrime(i))
                printf("Prime number %d (Rank %d ) \n", i, rank);
        }
    }

    MPI_Finalize();
    return 0;
}
```

```
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$ mpicc lab01_additional2.c -o lab01_additional2 && mpirun -np 2 ./lab01_additional2
Prime number 2 (Rank 0 )
Prime number 3 (Rank 0 )
Prime number 5 (Rank 0 )
Prime number 7 (Rank 0 )
Prime number 11 (Rank 0 )
Prime number 13 (Rank 0 )
Prime number 17 (Rank 0 )
Prime number 19 (Rank 0 )
Prime number 23 (Rank 0 )
Prime number 29 (Rank 0 )
Prime number 31 (Rank 0 )
Prime number 37 (Rank 0 )
Prime number 41 (Rank 0 )
Prime number 43 (Rank 0 )
Prime number 47 (Rank 0 )
Prime number 53 (Rank 1 )
Prime number 59 (Rank 1 )
Prime number 61 (Rank 1 )
Prime number 67 (Rank 1 )
Prime number 71 (Rank 1 )
Prime number 73 (Rank 1 )
Prime number 79 (Rank 1 )
Prime number 83 (Rank 1 )
Prime number 89 (Rank 1 )
Prime number 97 (Rank 1 )
Student@dblab-hp-29:~/Desktop/KaustavLABS4/PCAP LAB/LAB 01$
```

Ln 13, Col 26 Spaces: 4 UTF-8 LF C ROS