C hello.c

```c
#include <mpi.h>
#include <stdio.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("Hello World from process %d out of %d processes\n", rank, size);

    if(rank == 0) printf("Q0, Adarsh Ranjan 230962278");


    MPI_Finalize();

    return 0;
}

```

```
(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$ mpicc hello.c -o hello
(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$ mpirun -np 4 ./hello
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
(warning: it would break the library ABI, don't enable unless really needed).
Hello World from process 0 out of 4 processes
Hello World from process 1 out of 4 processes
Hello World from process 2 out of 4 processes
Hello World from process 3 out of 4 processes
Q0, Adarsh Ranjan 230962278(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$
```

C q1.c

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

long power(int base, int exp) {
    long res = 1;
    for (int i = 0; i < exp; i++) {
        res *= base;
    }
    return res;
}

int main(int argc, char *argv[])
{
    int rank, size;
    int x = 2;

    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    long result = power(x, rank);

    printf("Process %d: %d^%d = %ld\n", rank, x, rank, result);

    if (rank % 2 == 0)
        printf("Process %d says: Hello\n", rank);
    else
        printf("Process %d says: World\n", rank);


    if(rank == 0) {
        printf("Q1, Adarsh Ranjan 230962278\n");
    }

    MPI_Finalize();

    return 0;
}
```

```
(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$ mpirun -np 4 ./q1
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
(warning: it would break the library ABI, don't enable unless really needed).
Process 0: 2^0 = 1
Process 0 says: Hello
Q1, Adarsh Ranjan 230962278
Process 3: 2^3 = 8
Process 3 says: World
Process 1: 2^1 = 2
Process 1 says: World
Process 2: 2^2 = 4
Process 2 says: Hello
```

q2.c

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

int main(int argc, char *argv[])
{
    int rank, size;
    int a = 20, b = 10;

    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if (size < 4)
    {
        if (rank == 0)
            printf("Please run with at least 4 processes\n");
        MPI_Finalize();
        return 0;
    }

    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);

    if(rank == 0) printf("Q2,Adarsh Ranjan 230962278");


    MPI_Finalize();
    return 0;
}
```

```
(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$ mpicc q2.c -o q2
(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$ mpirun -np 4 ./q2
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
(warning: it would break the library ABI, don't enable unless really needed).
Subtraction: 20 - 10 = 10
Multiplication: 20 * 10 = 200
Division: 20 / 10 = 2
Addition: 20 + 10 = 30
Q2,Adarsh Ranjan 230962278(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$
```

C q3.c

```c
#include <mpi.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>

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

    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if (rank < len)
    {
        ch = str[rank];

        if (islower(ch))
            ch = toupper(ch);
        else
            ch = tolower(ch);
    }

    MPI_Gather(&ch, 1, MPI_CHAR,
               str, 1, MPI_CHAR,
               0, MPI_COMM_WORLD);

    if (rank == 0)
        printf("Toggled String: %s\n", str);

    if(rank == 0) printf("Q3, Adarsh Ranjan 230962278");


    MPI_Finalize();
    return 0;
}
```

```
(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$ mpicc q3.c -o q3
(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$ mpirun -np 5 ./q3
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
(warning: it would break the library ABI, don't enable unless really needed).
Toggled String: hello
Q3, Adarsh Ranjan 230962278(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$
```

C q4.c

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

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

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

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

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

    if (rank % 2 == 0)
        printf("Process %d (Even): Factorial = %lld\n", rank, factorial(rank));
    else
        printf("Process %d (Odd): Fibonacci = %lld\n", rank, fibonacci(rank));

    if(rank == 0) printf("Q4, Adarsh Ranjan 230962278");


    MPI_Finalize();
    return 0;
}
```

```
(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$ mpicc q4.c -o q4
(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$ mpirun -np 6 ./q4
hwloc/linux: Ignoring PCI device with non-16bit domain.
Pass --enable-32bits-pci-domain to configure to support such devices
(warning: it would break the library ABI, don't enable unless really needed).
Process 4 (Even): Factorial = 24
Process 2 (Even): Factorial = 2
Process 3 (Odd): Fibonacci = 2
Process 1 (Odd): Fibonacci = 1
Process 5 (Odd): Fibonacci = 5
Process 0 (Even): Factorial = 1
Q4, Adarsh Ranjan 230962278(base) mca@computinglab25-22:~/Desktop/PPL_230962278/Week1$
```