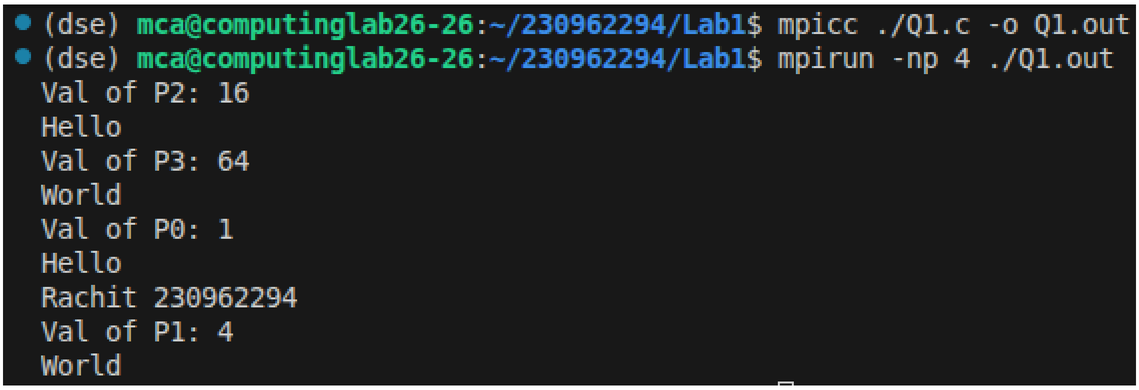
PPL Week-1

Rachit Takate – 230962294 – B54

I. Write a simple MPI program to find out pow (x, rank) for all the processes where ‘x’ is the integer constant and ‘rank’ is the rank of the process. Write a program in MPI where even ranked processes print “Hello” and odd ranked processes print “World”.

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

int powi(int x, int y) {
    int res = 1;
    for(int i = 0; i < y; i++)
        res *= x;

    return res;
}

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);

    printf("Val of P%d: %d\n", rank, powi(size, rank));

    if(rank % 2)
        printf("World\n");
    else
        printf("Hello\n");

    if(rank == 0)
        printf("Rachit 230962294\n");
    MPI_Finalize();
    return 0;
}
```

Output



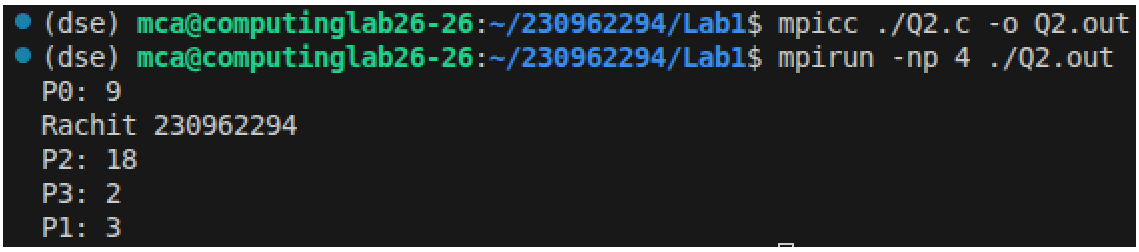
```
● (dse) mca@computinglab26-26:~/230962294/Lab1$ mpicc ./Q1.c -o Q1.out
● (dse) mca@computinglab26-26:~/230962294/Lab1$ mpirun -np 4 ./Q1.out
  Val of P2: 16
  Hello
  Val of P3: 64
  World
  Val of P0: 1
  Hello
  Rachit 230962294
  Val of P1: 4
  World
```

II. Write a program in MPI to simulate a simple calculator. Perform each operation using different processes in parallel.

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

```
int main(int argc, char* argv[]) {
    int a = 6, b = 3, res;
    int rank;
    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    switch (rank) {
    case 0:
        res = a + b;
        break;
    case 1:
        res = a - b;
        break;
    case 2:
        res = a * b;
        break;
    case 3:
        res = a / b;
        break;
    }

    printf("P%d: %d\n", rank, res);
    if(rank == 0)
        printf("Rachit 230962294\n");
    MPI_Finalize();
    return 0;
}
```

Output

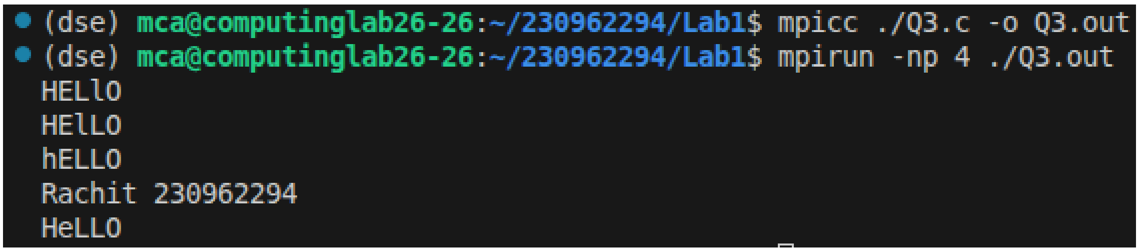


```
● (dse) mca@computinglab26-26:~/230962294/Lab1$ mpicc ./Q2.c -o Q2.out
● (dse) mca@computinglab26-26:~/230962294/Lab1$ mpirun -np 4 ./Q2.out
  P0: 9
  Rachit 230962294
  P2: 18
  P3: 2
  P1: 3
```

III. Write a program in MPI to toggle the character of a given string indexed by the rank of the process.

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

char inp[] = "HELLO";

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

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    if(inp[rank] > 96)
        inp[rank] -= 32;
    else
        inp[rank] += 32;

    printf("%s\n", inp);
    if(rank == 0)
        printf("Rachit 230962294\n");
    MPI_Finalize();
    return 0;
}
```

Output

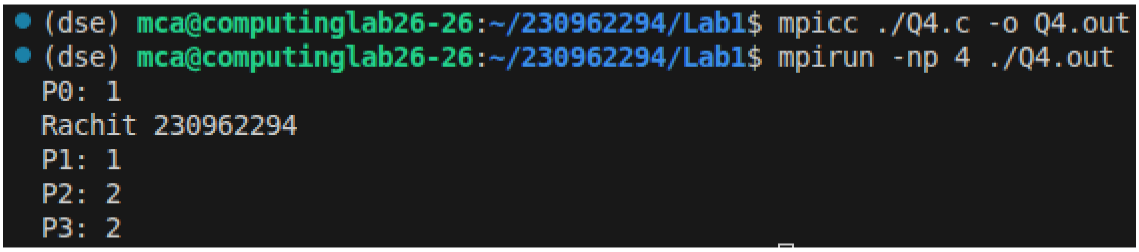


```
● (dse) mca@computinglab26-26:~/230962294/Lab1$ mpicc ./Q3.c -o Q3.out
● (dse) mca@computinglab26-26:~/230962294/Lab1$ mpirun -np 4 ./Q3.out
  HElLO
  HElLO
  hELLO
  Rachit 230962294
  HeLLO
```

IV. Write a program in MPI where even ranked processes print the factorial of the rank and odd ranked processes print the Fibonacci number of the rank.

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

int main(int argc, char* argv[]) {
    int rank, res = 1;
    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    if(rank % 2) {
        if(rank < 3)
            res = rank;
        else {
            int a = 0, b = 1;

            for(int i = 2; i <= rank; i++) {
                res = a + b;
                a = b;
                b = res;
            }
        }
    }
    else {
        for(int i = 2; i <= rank; i++)
            res *= i;
    }

    printf("P%d: %d\n", rank, res);

    if(rank == 0)
        printf("Rachit 230962294\n");
    MPI_Finalize();
    return 0;
}
```

Output



```
● (dse) mca@computinglab26-26:~/230962294/Lab1$ mpicc ./Q4.c -o Q4.out
● (dse) mca@computinglab26-26:~/230962294/Lab1$ mpirun -np 4 ./Q4.out
  P0: 1
  Rachit 230962294
  P1: 1
  P2: 2
  P3: 2
```