Aryaman Mishra

19BCE1027

LAB 6

1. **Simple hello world mpi**

```
#include<stdio.h>

#include <mpi.h>

int main()

{

printf("hello world\n");

return 0;

}
```

```
/home/mobaxterm/Desktop/19BCE1027_PCD
29/09/2021  11:10.46  mpicc -o a lab61.c


/home/mobaxterm/Desktop/19BCE1027_PCD
29/09/2021  11:10.54  mpirun a.exe
hello world
hello world
hello world
hello world


/home/mobaxterm/Desktop/19BCE1027_PCD
29/09/2021  11:11.20  mpirun -np 3 ./a
hello world
hello world
hello world
```

2. **Getting started with MPI program**

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

int main(int argc, char** argv) {
    // Initialize the MPI environment
    MPI_Init(NULL, NULL);
```

```
    // Get the number of processes
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    // Get the rank of the process
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    // Get the name of the processor
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    // Print off a hello world message
    printf("Hello world from processor %s, rank %d out of %d processors\n",
           processor_name, world_rank, world_size);

    // Finalize the MPI environment.
    MPI_Finalize();
    return 0;
}
```

```
/home/mobaxterm/Desktop/19BCE1027_PCD
29/09/2021  11:11.51  mpicc -o a lab62.c


/home/mobaxterm/Desktop/19BCE1027_PCD
29/09/2021  11:12.51  ./a
Hello world from processor LAPTOP-B45MQN7C, rank 0 out of 1 processors


/home/mobaxterm/Desktop/19BCE1027_PCD
29/09/2021  11:13.04  mpirun -np 4 ./a
Hello world from processor LAPTOP-B45MQN7C, rank 0 out of 4 processors
Hello world from processor LAPTOP-B45MQN7C, rank 1 out of 4 processors
Hello world from processor LAPTOP-B45MQN7C, rank 2 out of 4 processors
Hello world from processor LAPTOP-B45MQN7C, rank 3 out of 4 processors
```

3. **Write a MPI program that will print your name along with all processes with odd ranks.**

```
#include <mpi.h>

#include <stdio.h>



int main(int argc, char** argv) {
```

```
    // Initialize the MPI environment
    MPI_Init(NULL, NULL);

    // Get the number of processes
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    // Get the rank of the process
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    // Get the name of the processor
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    // Print off a hello world message
    //printf("Hello world from processor %s, rank %d out of %d processors\n",
          //processor_name, world_rank, world_size);
    if(world_rank%2==0)
    {
           printf("Hello world from processor %s, rank %d out of %d
processors\n",processor_name, world_rank, world_size);
           }
           else
           {
```

```
                printf("Aryaman Mishra 19BCE1027 from processor %s, rank %d out of %d processors\n",processor_name, world_rank, world_size);

        }

    // Finalize the MPI environment.

    MPI_Finalize();

    return 0;

}
```