Name: Hiten Raj Singh

CSE C, 230905152

Roll No. 23

Q1a.

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

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);
double result = pow(x, rank);
printf("Process %d pow(%d, %d) = %.2f\n", rank, x, rank, result);
MPI_Finalize();
return 0;
}
```

Screenshot:

```
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpicc -o abc q1a.c -lm
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpirun -np 4 ./abc
Process 3 pow(2, 3) = 8.00
Process 0 pow(2, 0) = 1.00
Process 1 pow(2, 1) = 2.00
Process 2 pow(2, 2) = 4.00
```

Q1b.

```c
#include <stdio.h>
#include <mpi.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("Process %d replied: Hello\n", rank);
} else {
printf("Process %d replied: World\n", rank);
}
MPI_Finalize();
return 0;
}
```

Screenshot:

```
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpicc -o abc q1b.c
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpirun -np 4 ./abc
Process 2 replied: Hello
Process 0 replied: Hello
Process 1 replied: World
Process 3 replied: World
```

Q2.

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

int main(int argc, char *argv[]) {
int rank, size;
int a = 24, b = 6;
printf("a: %d, b: %d , ", a, b);
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);
if(rank%4 == 0) printf("a + b = %d\n", a+b);
else if(rank%4 == 1) printf("a - b = %d\n", a-b);
else if(rank%4 == 2) printf("a * b = %d\n", a*b);
else printf("a / b = %d\n", a/b);
MPI_Finalize();
return 0;
}
```

Screenshot:

```
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpicc -o abc q2.c
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpirun -np 4 ./abc
a: 24, b: 6  ,  a * b = 144
a: 24, b: 6  ,  a / b = 4
a: 24, b: 6  ,  a + b = 30
a: 24, b: 6  ,  a - b = 18
```

Q3.

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

int main(int argc, char *argv[]) {
int rank, size;
char word[] = "HeLlO";
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);
if (rank < sizeof(word) - 1) {
char before= word[rank];
if (word[rank]>=65 && word[rank]<=90) {
word[rank] = word[rank]+32;
} else {
word[rank] = word[rank]-32;
}
printf("Process %d toggled character: %c to %c.\n", rank, before, word[rank]);
}
MPI_Finalize();
return 0;
}
```

Screenshot:

```
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpicc -o abc q3.c
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpirun -np 5 ./abc
Process 1 toggled character: e to E.
Process 2 toggled character: L to l.
Process 3 toggled character: l to L.
Process 4 toggled character: O to o.
Process 0 toggled character: H to h.
```

Q4.

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

```c
#include <mpi.h>
long fact(int n) {
long fact = 1;
for (int i = 1; i <= n; i++) {
fact *= i;
}
return fact;
}
long fibo(int n) {
if (n <= 1) return n;
long a = 0, b = 1, temp;
for (int i = 2; i <= n; i++) {
temp = a + b;
a = b;
b = temp;
}
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("Process %d Factorial is: %ld\n", rank, fact(rank));
} else {
printf("Process %d Fibonacci is: %ld\n", rank, fibo(rank));
}
MPI_Finalize();
return 0;
}
```

Screenshot:

```
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpicc -o abc q4.c
6CSEC1@debian:~/Documents/230905152_PPL/L1$ mpirun -np 6 ./abc
Process 2 Factorial is: 2
Process 5 Fibonacci is: 5
Process 4 Factorial is: 24
Process 0 Factorial is: 1
Process 1 Fibonacci is: 1
Process 3 Fibonacci is: 2
```

QA1

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

int reverse (int n) {
    int rev = 0;
    while (n > 0) {
        rev = rev*10 + n%10;
        n = n/10;
    }
    return rev;
}

int main (int argc, char *argv[]) {
    int rank, size;
    int arr[] = {18, 523, 301, 1234, 2, 14, 108, 150, 1928};

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

    printf("[%d] %d\n", rank, reverse(arr[rank]));

    MPI_Finalize();
    return 0;
}
```

QA2.

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

int isPrime (int n) {
    if (n <= 1) return 0;
    for (int i = 2; i < n; i++)
        if (n % i == 0) return 0;
    return 1;
}

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

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
```

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

    printf("[%d]", rank);
    for (int i = 50*rank+1; i <= 50*(rank+1); i++)
        if (isPrime(i)) printf (" %d", i);
    printf("\n");

    MPI_Finalize();
    return 0;
}
```

Screenshot:

```
6CSE1@debian:/Desktop/230905152/lab1$ mpirun -n 9
[3] 4321
[5] 41
[8] 8291
[6] 801
[7] 51
[4] 103
[1] 81
[1] 325
6CSE1@debian:/Desktop/230905152/lab1$ mpirun -n 2
[1] 53 59 61 67 71 73 79 83 89 89 97
[0] 2 3 5 7 11 13 17 19 23 29 31 37 41 44 43 47
```