PP LAB WEEK-6

DSE VI-A2 Divansh Prasad 210968140

1) Write a simple MPI program to find out pow(x,rank) for all processes where ‘x’ is the integer constant and ‘rank’ is the rank of the process.

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

int main(int argc, char *argv[]) {
   int rank, size, x, result;
   MPI_Init(&argc, &argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);
   MPI_Comm_size(MPI_COMM_WORLD, &size);

   if (rank == 0) {
       printf("\nEnter the value of x: ");
       scanf("%d", &x);
   }

   MPI_Bcast(&x, 1, MPI_INT, 0, MPI_COMM_WORLD);

   result = pow(x, rank);

   int *results = NULL;
   if (rank == 0) {
       results = (int *)malloc(size * sizeof(int));
   }
   MPI_Gather(&result, 1, MPI_INT, results, 1, MPI_INT, 0,
MPI_COMM_WORLD);

   if (rank == 0) {
       printf("\nResults:\n");
       for (int i = 0; i < size; i++) {
           printf("Process %d: %d\n", i, results[i]);
       }
       free(results);
```

```
    }

    MPI_Finalize();
    return 0;
}
```

```
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpic++ -o PowerRank PowerRank.cpp
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 8 ./PowerRank

7
Enter the value of x:
Results:
Process 0: 1
Process 1: 7
Process 2: 49
Process 3: 343
Process 4: 2401
Process 5: 16807
Process 6: 117649
Process 7: 823543
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 5 ./PowerRank

50
Enter the value of x:
Results:
Process 0: 1
Process 1: 50
Process 2: 2500
Process 3: 125000
Process 4: 6250000
```

2) Write a program in MPI where even ranked process prints “Hello” and odd ranked process prints “World”.

```
#include<stdio.h>
#include<mpi.h>
#include<stdlib.h>
int main(int argc, char* argv[])
{
   int myid, numprocs=10, namelen;
   char processor_name[MPI_MAX_PROCESSOR_NAME];
   MPI_Init(&argc, &argv);         // starts MPI
   MPI_Comm_rank(MPI_COMM_WORLD, &myid);  // get current process id
   MPI_Comm_size(MPI_COMM_WORLD, &numprocs);       // get number of
processeser
   MPI_Get_processor_name(processor_name, &namelen);
   printf("Rank %d: ", myid);
   if (myid % 2 == 0) {
       printf("Hello\n");
   }
   else {
       printf("World\n");
   }
   MPI_Finalize();
   return 0;
}
```

```
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpic++ -o EvenOdd EvenOdd.cpp
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 8 ./EvenOdd
Rank 0: Hello
Rank 0: World
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpic++ -o EvenOdd EvenOdd.cpp
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 8 ./EvenOdd
Rank 5: World
Rank 1: World
Rank 4: Hello
Rank 7: World
Rank 3: World
Rank 0: Hello
Rank 6: Hello
Rank 2: Hello
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 4 ./EvenOdd
Rank 0: Hello
Rank 1: World
Rank 2: Hello
Rank 3: World
```

3) Write a program in MPi to simulate a simple calculator. Perform each operation using a different thread.

```c
#include <stdio.h>
#include <mpi.h>
#include <stdlib.h>
#include <math.h>
int main(int argc, char* argv[]) {
   int myid, numprocs, namelen, num1 = 69, num2 = 23;
   char processor_name[MPI_MAX_PROCESSOR_NAME];
   MPI_Init(&argc, &argv); // starts MPI
   MPI_Comm_rank(MPI_COMM_WORLD, &myid); // get current process id
   MPI_Comm_size(MPI_COMM_WORLD, &numprocs); // get number of processes
   MPI_Get_processor_name(processor_name, &namelen);
   if (myid % 4 == 0)
       printf("Sum: %d\n", num1 + num2);
   if (myid % 4 == 1)
       printf("Difference: %d\n", num1 - num2);
   if (myid % 4 == 2)
       printf("Product: %d\n", num1 * num2);
   if (myid % 4 == 3)
       printf("Division: %d\n", num1 / num2);
   MPI_Finalize();
   return 0;
}
```

```
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpic++ -o Calculator Calculator.cpp
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 4 ./Calculator
Division: 3
Sum: 92
Product: 1587
Difference: 46
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 6 ./Calculator
Sum: 92
Product: 1587
Difference: 46
Sum: 92
Difference: 46
Division: 3
```

4) Write a program in MPI to toggle the characters of a given string indexed by the rank of the process.

```c
#include <stdio.h>
#include <mpi.h>
#include <stdlib.h>
#include <ctype.h>
int main(int argc, char* argv[]) {
   int myid, numprocs, namelen, len = 7;
   char s[7] = "HeLlO";
   char processor_name[MPI_MAX_PROCESSOR_NAME];

   MPI_Init(&argc, &argv); // starts MPI
   MPI_Comm_rank(MPI_COMM_WORLD, &myid); // get current process id
   MPI_Comm_size(MPI_COMM_WORLD, &numprocs); // get number of processes
   MPI_Get_processor_name(processor_name, &namelen);

   if (myid < len) {
       if (islower(s[myid])) {
           s[myid] = toupper(s[myid]);
       } else if (isupper(s[myid])) {
           s[myid] = tolower(s[myid]);
       }
   }
   MPI_Finalize();
```

```
    printf("%s\n", s);
    return 0;
}
```

```
divansh@ROG-STRIX:~/PP-Lab/Week-6$ echo "Input String: "HeLlO""
Input String: HeLlO
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpic++ -o ToggleString ToggleString.cpp
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 5 ./ToggleString
HELlO
HellO
heLlO
HeLLO
HeLlo
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 1 ./ToggleString
heLlO
divansh@ROG-STRIX:~/PP-Lab/Week-6$ mpiexec -n 2 ./ToggleString
HELlO
heLlO
```