

PPL WEEK1 SUBMISSION

Name: Davasam Karthikeya Section: AIMLB Reg No: 230962326

1. 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 process prints “Hello” and odd ranked process prints “World”.

Code:-

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

int int_pow(int x, int power){
    int ans = 1;
    while(power >= 1){
        ans *= x;
        power -= 1;
    }
    return ans;
}

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

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

    if(rank == 0)
        printf("-----Davasam Karthikeya, 230962326-----\n");

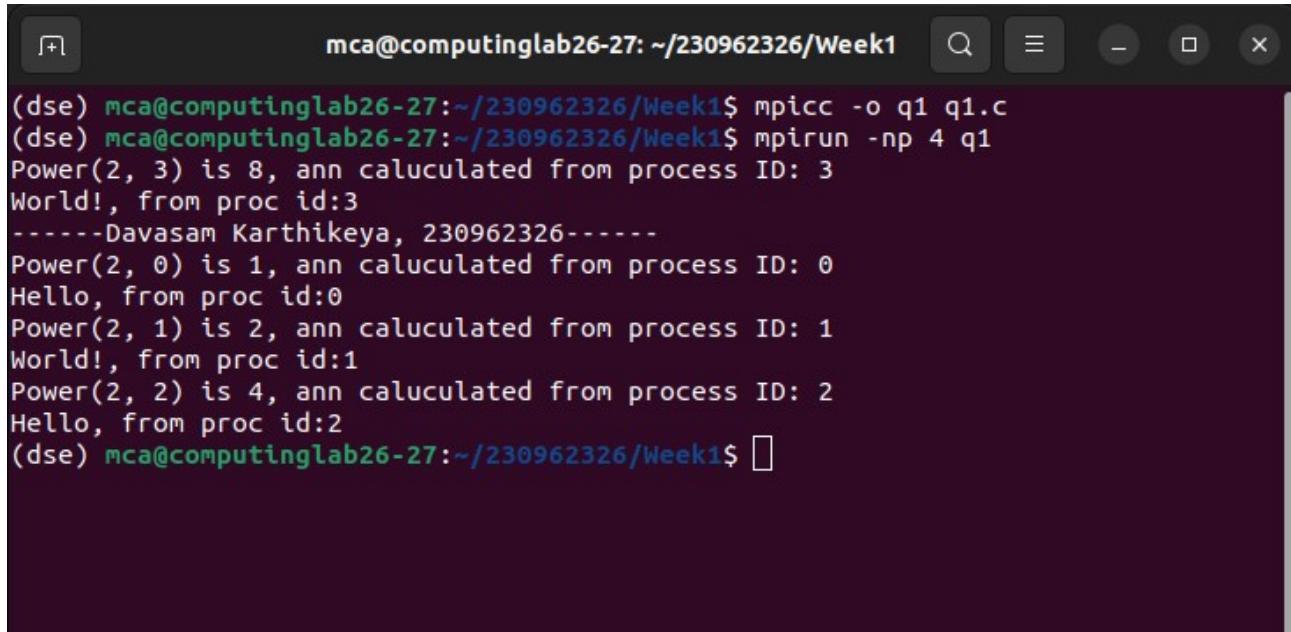
    printf("Power(%d, %d) is %d, ann caluculated from process ID: %d \n", x,
    rank, int_pow(x, rank), rank);

    if(rank % 2 == 0){
        // even ranked proc
        printf("Hello, from proc id:%d \n", rank);
    }else{
        printf("World!, from proc id:%d \n", rank);
    }

    MPI_Finalize();

    return 0;
}
```

Output:-



```
mca@mca@computinglab26-27:~/230962326/Week1$ mpicc -o q1 q1.c
mca@mca@computinglab26-27:~/230962326/Week1$ mpirun -np 4 q1
Power(2, 3) is 8, ann caluculated from process ID: 3
World!, from proc id:3
-----Davasam Karthikeya, 230962326-----
Power(2, 0) is 1, ann caluculated from process ID: 0
Hello, from proc id:0
Power(2, 1) is 2, ann caluculated from process ID: 1
World!, from proc id:1
Power(2, 2) is 4, ann caluculated from process ID: 2
Hello, from proc id:2
(mca@mca@computinglab26-27:~/230962326/Week1$ )
```

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

Code:-

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

int main(int argc, char *argv[]){
    int x = 2, y = 3;

    int rank, size;
    MPI_Init(&argc, &argv);

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

    if(rank == 0)
        printf("-----Davasam Karthikeya, 230962326-----\n");

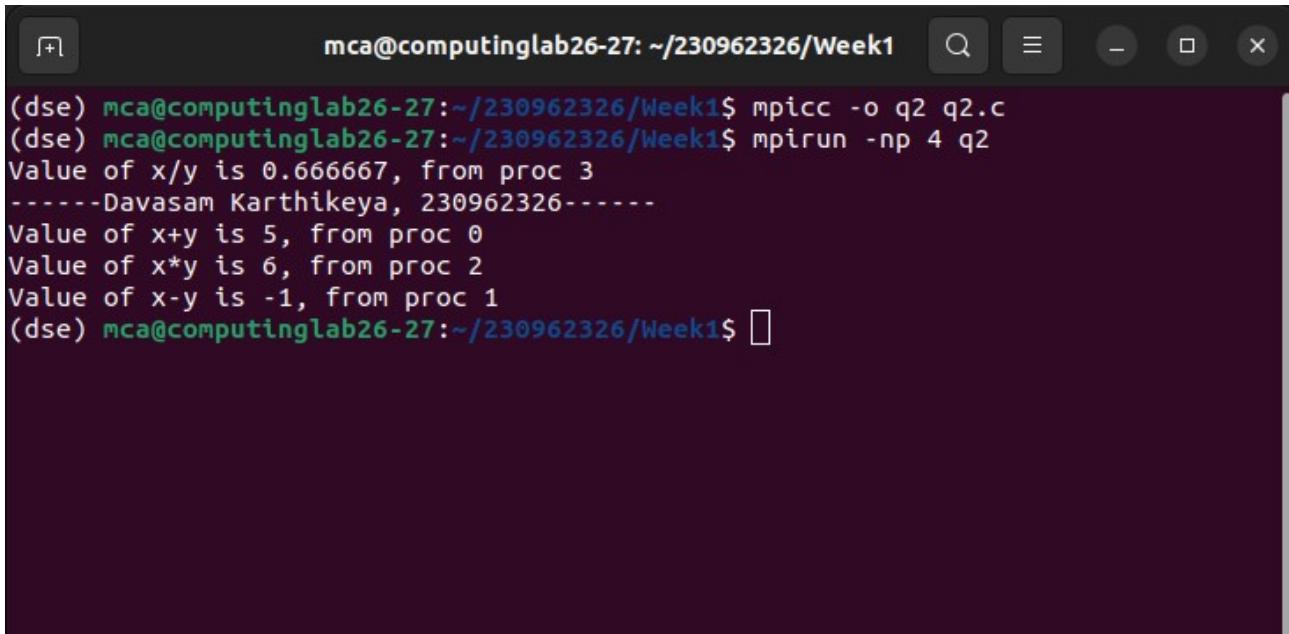
    switch(rank){
        case 0:
            printf("Value of x+y is %d, from proc 0\n", x+y);
            break;
        case 1:
            printf("Value of x-y is %d, from proc 1\n", x-y);
            break;
        case 2:
            printf("Value of x*y is %d, from proc 2\n", x*y);
            break;
        case 3:
            printf("Value of x/y is %f, from proc 3\n", ((float)x)/y);
            break;
    }
}
```

```

        MPI_Finalize();
}

```

Output:-



A screenshot of a terminal window titled "mca@computinglab26-27: ~/230962326/Week1". The window contains the following text:

```

(dse) mca@computinglab26-27:~/230962326/Week1$ mpicc -o q2 q2.c
(dse) mca@computinglab26-27:~/230962326/Week1$ mpirun -np 4 q2
Value of x/y is 0.666667, from proc 3
-----Davasam Karthikeya, 230962326-----
Value of x+y is 5, from proc 0
Value of x*y is 6, from proc 2
Value of x-y is -1, from proc 1
(dse) mca@computinglab26-27:~/230962326/Week1$ 

```

3. write a program in MPI to toggle the character of a given string indexed by the rank of the process. Hint: Suppose the string is HELLO and there are 5 processes, then process 0 toggle “H” to “h”, process 1 toggle ‘E’ to ‘e’ and so on.

Code:-

```

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

int main(int argc, char *argv[]){
    // ques cosnts
    int arr_size = 5;
    char word[] = "HELLO";

    int rank, size;
    MPI_Init(&argc, &argv);

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

    if(rank == 0)
        printf("-----Davasam Karthikeya, 230962326-----\n");

    if(rank < arr_size){
        char prev = word[rank];
        word[rank] = (word[rank] < 91) ? word[rank]+32: word[rank] - 32;
        printf("In ProcID: %d, letter is toggled from %c to %c \n", rank, prev,
        word[rank]);
    }
}

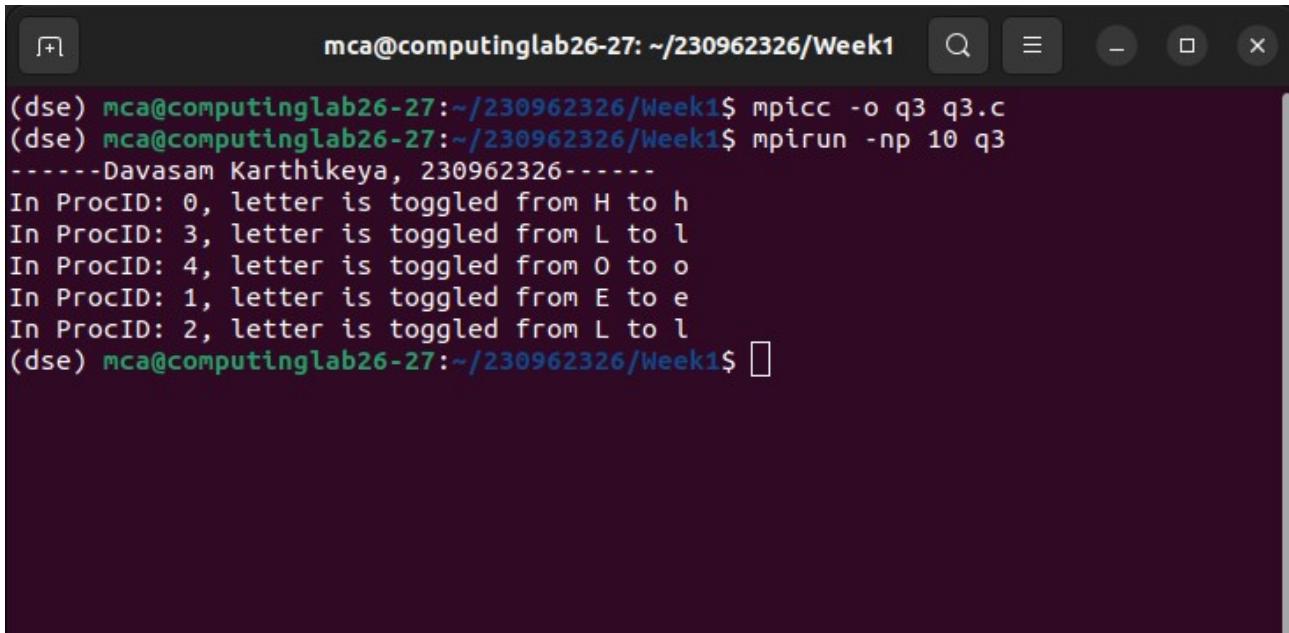
```

```

MPI_Finalize();
}

```

Output:-



A terminal window titled "mca@computinglab26-27: ~/230962326/Week1". The command mpicc -o q3 q3.c was run, followed by mpirun -np 10 q3. The output shows 10 parallel processes printing their rank and toggled letter. The first process (rank 0) prints "In ProcID: 0, letter is toggled from H to h". Subsequent processes print similar messages for ranks 1 through 9. The last process (rank 9) prints "In ProcID: 9, letter is toggled from Z to z".

```

mca@mca@computinglab26-27:~/230962326/Week1$ mpicc -o q3 q3.c
mca@mca@computinglab26-27:~/230962326/Week1$ mpirun -np 10 q3
-----Davasam Karthikeya, 230962326-----
In ProcID: 0, letter is toggled from H to h
In ProcID: 1, letter is toggled from E to e
In ProcID: 2, letter is toggled from L to l
In ProcID: 3, letter is toggled from L to l
In ProcID: 4, letter is toggled from O to o
In ProcID: 5, letter is toggled from S to s
In ProcID: 6, letter is toggled from T to t
In ProcID: 7, letter is toggled from U to u
In ProcID: 8, letter is toggled from V to v
In ProcID: 9, letter is toggled from Z to z
mca@mca@computinglab26-27:~/230962326/Week1$ 

```

4. Write a program in MPI where even ranked process prints factorial of the rank and odd ranked process prints ranks Fibonacci number.

Code:-

```

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

int fact_number(int x){
    int ans = 1;
    while(x > 0){
        ans *= x;
        x--;
    }
    return ans;
}

int nthFibonacci(int n){
    if (n <= 1) return n;
    return nthFibonacci(n - 1) + nthFibonacci(n - 2);
}

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

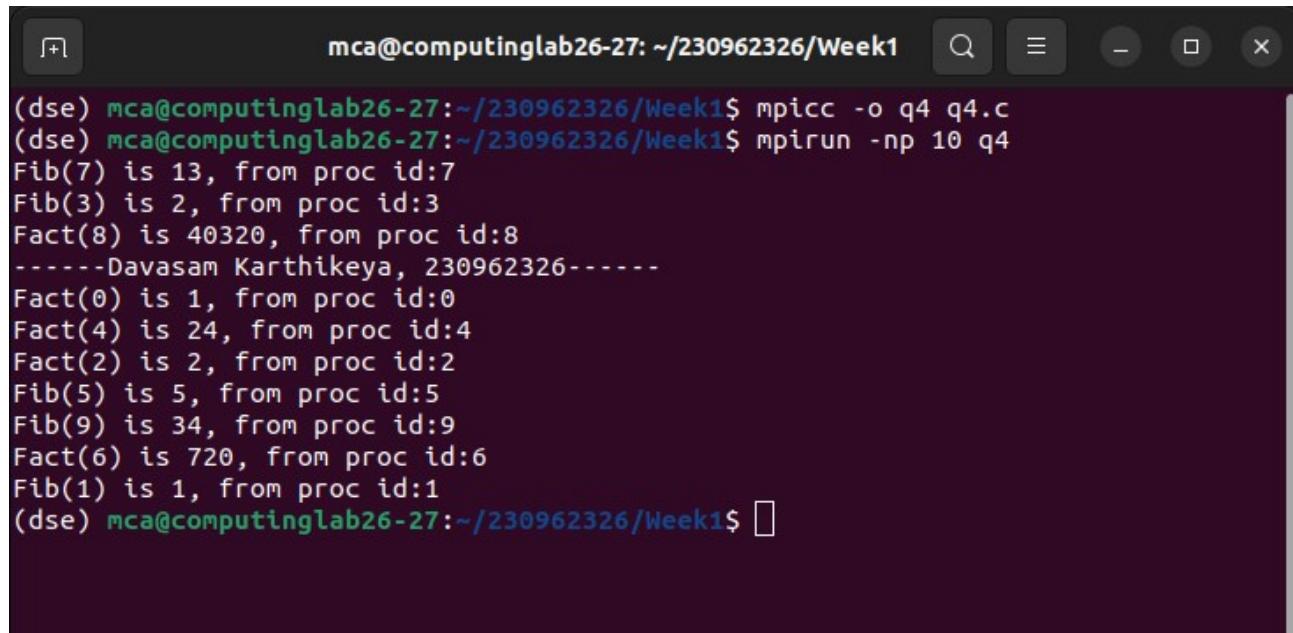
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if(rank == 0)
        printf("-----Davasam Karthikeya, 230962326-----\n");

```

```
if(rank % 2 == 0)
    printf("Fact(%d) is %d, from proc id:%d\n", rank, fact_number(rank),
rank);
else
    printf("Fib(%d) is %d, from proc id:%d\n", rank, nthFibonacci(rank),
rank);
MPI_Finalize();
}
```

Output:-

A screenshot of a terminal window titled "mca@computinglab26-27: ~/230962326/Week1". The window shows the execution of an MPI program. The user first runs "mpicc -o q4 q4.c" to compile the code. Then, they run "mpirun -np 10 q4" to execute it with 10 processes. The output displays the results of both Fact and Fib functions for ranks 0 through 9, with each result including the rank, function name, value, and process ID. A personal note "-----Davasam Karthikeya, 230962326-----" is also present.

```
(dse) mca@computinglab26-27:~/230962326/Week1$ mpicc -o q4 q4.c
(dse) mca@computinglab26-27:~/230962326/Week1$ mpirun -np 10 q4
Fib(7) is 13, from proc id:7
Fib(3) is 2, from proc id:3
Fact(8) is 40320, from proc id:8
-----Davasam Karthikeya, 230962326-----
Fact(0) is 1, from proc id:0
Fact(4) is 24, from proc id:4
Fact(2) is 2, from proc id:2
Fib(5) is 5, from proc id:5
Fib(9) is 34, from proc id:9
Fact(6) is 720, from proc id:6
Fib(1) is 1, from proc id:1
(dse) mca@computinglab26-27:~/230962326/Week1$ 
```