Date : 22/01/2026

PPL WEEK3 SUBMISSION

Name: Davasam Karthikeya   Section: AIMLB   Reg No: 230962326

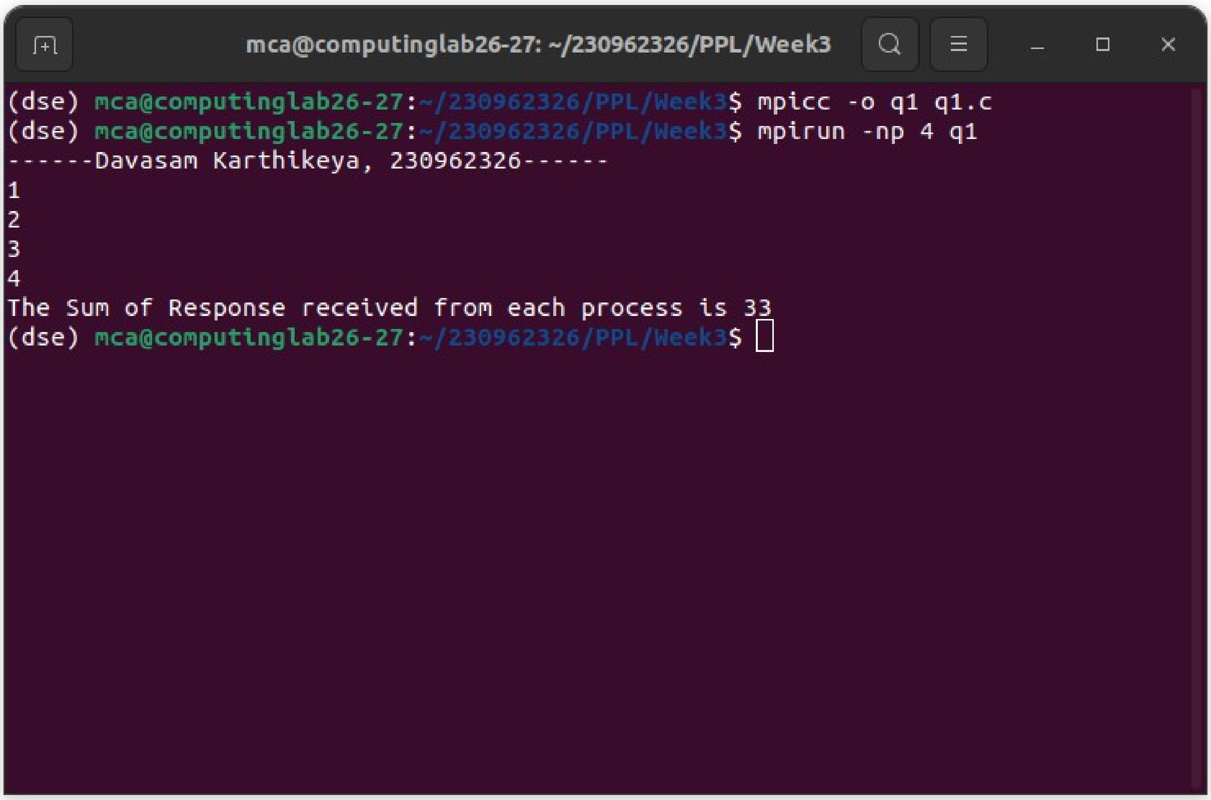
1. Write a MPI program to read N values in the root process. Root process sends one value to each process. Every process receives it and finds the factorial of that number and returns it to the root process. Root process gathers the factorial and finds sum of it. Use N number of processes.

Code:-

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

int fact_num(int n){
    if(n <= 1) return 1;
    return n * fact_num(n-1);
}

int main(int argc, char *argv[]){
    int size, rank, *arr, recvBuf;
    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");
        arr = (int*) calloc(size, sizeof(int));
        for(int i = 0; i< size; i++)scanf("%d", arr+i);
    }

    MPI_Scatter(arr, 1, MPI_INT, &recvBuf, 1, MPI_INT, 0, MPI_COMM_WORLD);
    recvBuf = fact_num(recvBuf);

    MPI_Gather(&recvBuf, 1, MPI_INT, arr, 1, MPI_INT, 0, MPI_COMM_WORLD);

    if(rank == 0){
        int sum = 0;
        for(int i =0; i< size; i++)sum += arr[i];
        printf("The Sum of Response received from each process is %d \n", sum);
    }

    MPI_Finalize();
    return 0;
}
```

Date : 22/01/2026

Output:-



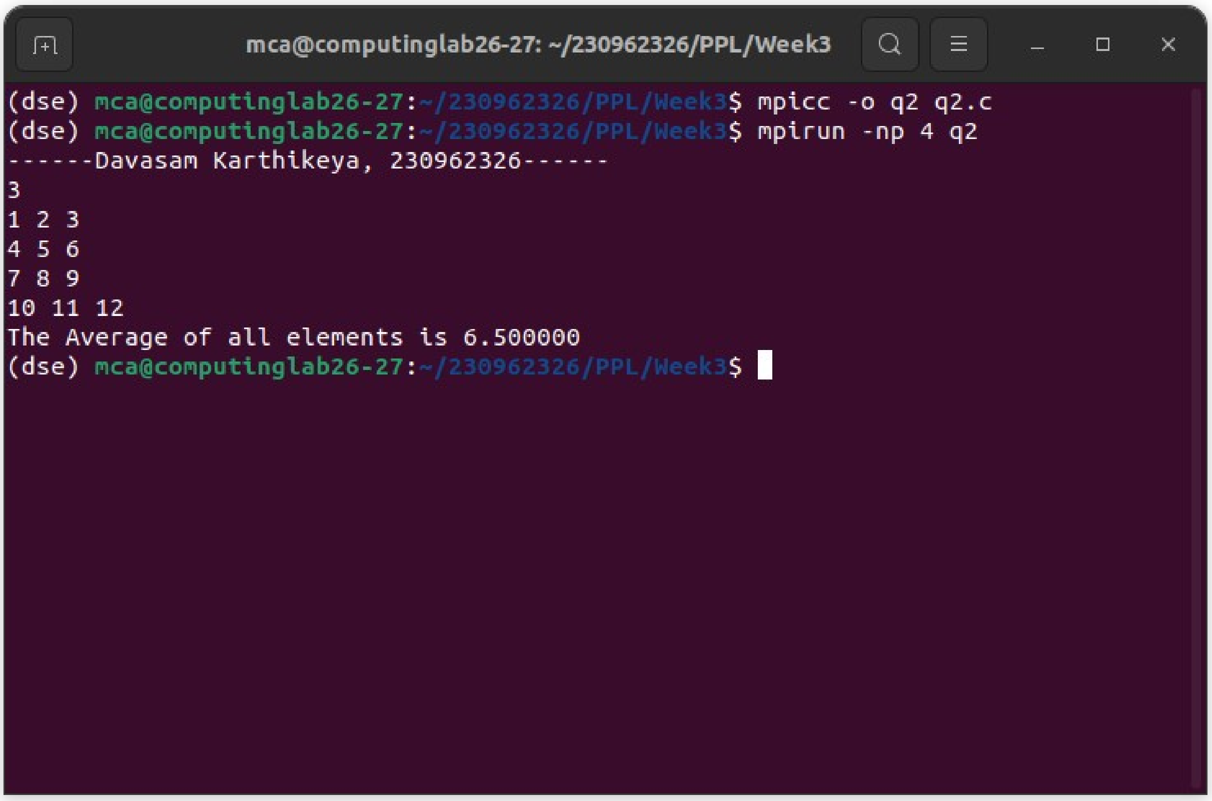
```
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ mpicc -o q1 q1.c
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ mpirun -np 4 q1
------Davasam Karthikeya, 230962326------
1
2
3
4
The Sum of Response received from each process is 33
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ 
```

2. Write a MPI program to read an integer value M and NXM elements into an 1D array in the root process, where N is the number of processes. Root process sends M elements to each process. Each process finds average of M elements it received and sends these average values to root. Root collects all the values and finds the total average. Use collective com- munication routines.

Code:-

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

int main(int argc, char *argv[]){
    int size, rank, *arr, m, *recvBuf;
    float *recvArr;
    MPI_Init(&argc, &argv);

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

    if(rank == 0){
        printf("------Davasam Karthikeya, 230962326------\n");
        scanf("%d", &m);
        arr = (int*)calloc(m*size, sizeof(int));
        for(int i = 0; i < m*size; i++)scanf("%d", arr + i);
        recvArr = (float*) calloc(size, sizeof(float));
    }
    MPI_Bcast(&m, 1, MPI_INT, 0, MPI_COMM_WORLD);
    recvBuf = (int*) calloc(m, sizeof(int));
    MPI_Scatter(arr, m, MPI_INT, recvBuf, m, MPI_INT, 0, MPI_COMM_WORLD);
    float sum = 0.0;
    for(int i =0; i<m; i++)sum += recvBuf[i];
    float avg_proc = sum / m;

    MPI_Gather(&avg_proc, 1, MPI_FLOAT, recvArr, 1, MPI_FLOAT, 0,
MPI_COMM_WORLD);

    if(rank == 0){
        sum = 0.0;
        for(int i =0; i<size; i++)sum += recvArr[i];
        printf("The Average of all elements is %f \n", sum/size);
    }

    MPI_Finalize();
    return 0;
}
```

Output:-



```
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ mpicc -o q2 q2.c
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ mpirun -np 4 q2
------Davasam Karthikeya, 230962326------
3
1 2 3
4 5 6
7 8 9
10 11 12
The Average of all elements is 6.500000
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$
```

Date : 22/01/2026

3. Write a MPI program to read a string. Using N processes (string length is evenly divisible by N), find the number of non-vowels in the string. In the root process print number of non-vowels found by each process and print the total number of non-vowels.

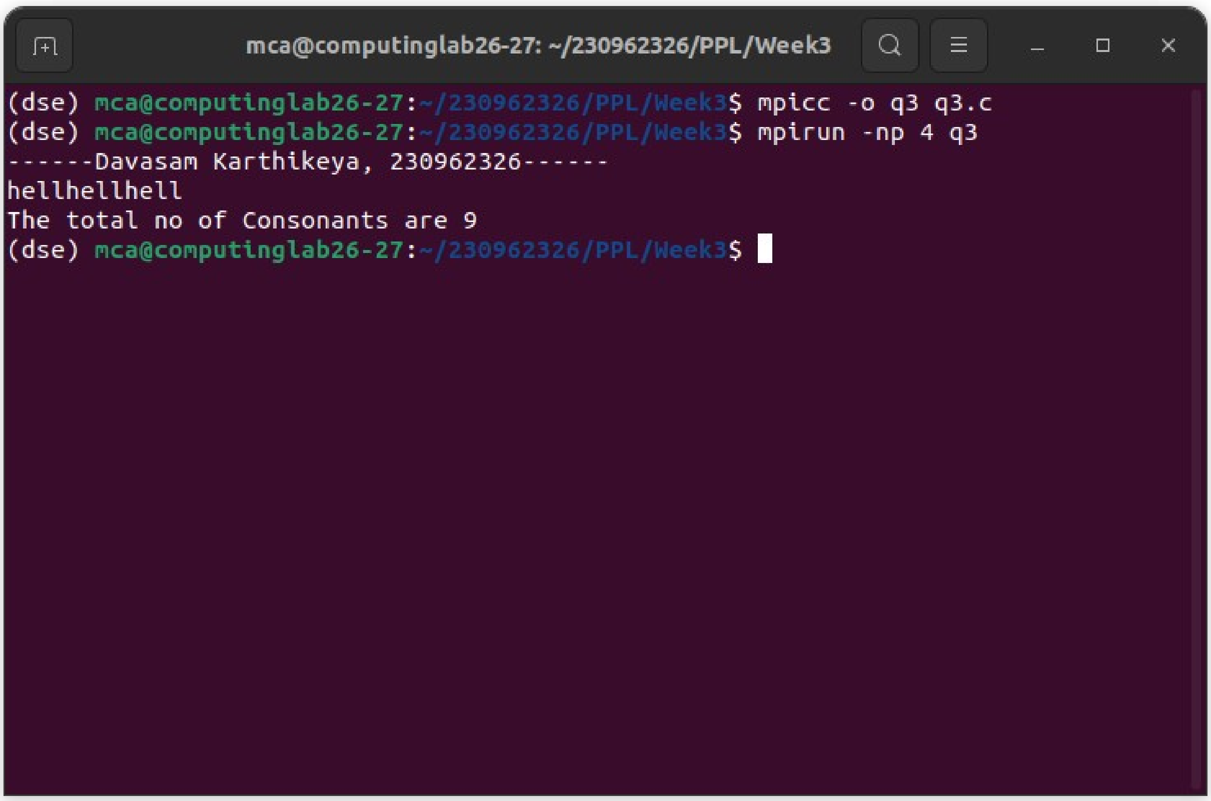
Code:-

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

int isConsonant(char ch){
    switch(ch){
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
            return 0;
        break;
    }
    return 1;
}

int main(int argc, char *argv[]){
    int size, rank, n;
    char str1[100];

    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");
        scanf("%s", str1);
        n = strlen(str1);
    }
    MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);

    if(n % size != 0){
        if(rank == 0)printf("The len of string is not a Multiple of N \n");
        MPI_Finalize();
        return 0;
    }
    n /= size;

    char recvStr[n+1];
    MPI_Scatter(str1, n, MPI_CHAR, recvStr, n, MPI_CHAR, 0, MPI_COMM_WORLD);
    recvStr[n] = '\0';

    int Conscount = 0;
    for(int i =0; i < n; i++)Conscount += isConsonant(recvStr[i]);

    int recvCount[size];
    MPI_Gather(&Conscount, 1, MPI_INT, recvCount, 1, MPI_INT, 0,
MPI_COMM_WORLD);

    if(rank == 0){
```

Date : 22/01/2026

```
        int sum = 0;
        for(int i = 0;i < size; i++)sum += recvCount[i];
        printf("The total no of Consonants are %d \n", sum);
    }

    MPI_Finalize();
    return 0;
}
```

Output:-



```
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ mpicc -o q3 q3.c
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ mpirun -np 4 q3
------Davasam Karthikeya, 230962326------
hellhellhell
The total no of Consonants are 9
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ 
```

4. Write a MPI Program to read two strings S1 and S2 of same length in the root process. Using N processes including the root (string length is evenly divisible by N), produce the resultant string as shown below. Display the resultant string in the root process. Use Col- lective Communication routines. Example: String S1: string String S2: length Resultant String : slterningtgh

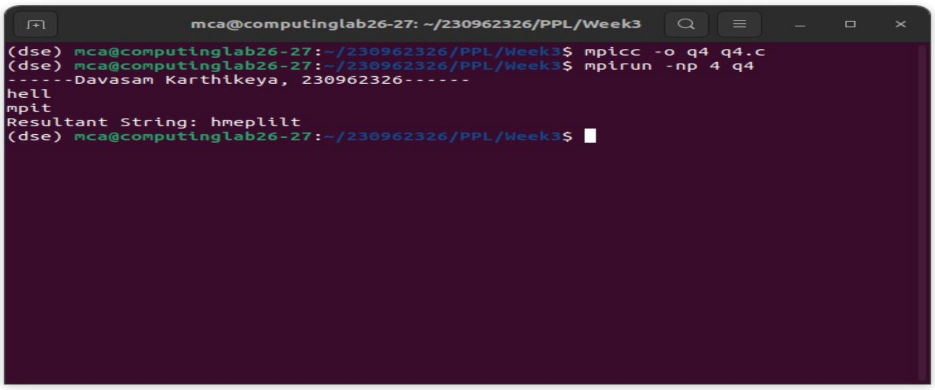
Code:-

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

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

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
```

Date : 22/01/2026

```
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    char S1[100], S2[100];

    if(rank == 0){
        printf("------Davasam Karthikeya, 230962326------\n");
        scanf("%s", S1);
        scanf("%s", S2);
        len = strlen(S1);
    }

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

    if(len % size != 0){
        MPI_Finalize();
        return 0;
    }

    chunk = len / size;

    char localS1[chunk], localS2[chunk], localResult[2 * chunk];

    MPI_Scatter(S1, chunk, MPI_CHAR, localS1, chunk, MPI_CHAR, 0,
MPI_COMM_WORLD);
    MPI_Scatter(S2, chunk, MPI_CHAR, localS2, chunk, MPI_CHAR, 0,
MPI_COMM_WORLD);

    for(int i = 0, j = 0; i < chunk; i++){
        localResult[j++] = localS1[i];
        localResult[j++] = localS2[i];
    }

    char result[2 * len + 1];

    MPI_Gather(localResult, 2 * chunk, MPI_CHAR, result, 2 * chunk, MPI_CHAR, 0,
MPI_COMM_WORLD);

    if(rank == 0){
        result[2 * len] = '\0';
        printf("Resultant String: %s\n", result);
    }

    MPI_Finalize();
    return 0;
}
```

Output:-



```
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ mpicc -o q4 q4.c
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$ mpirun -np 4 q4
------Davasam Karthikeya, 230962326------
hell
mpit
Resultant String: hmeplilt
(dse) mca@computinglab26-27:~/230962326/PPL/Week3$
```