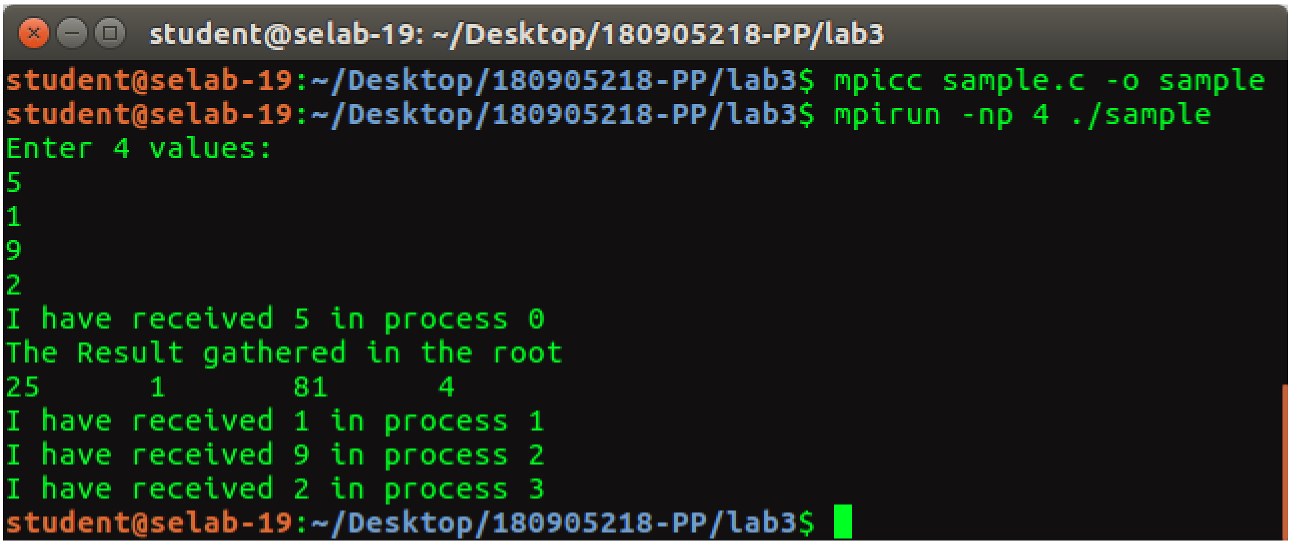
NAME: Arunima Singh Thakur
SECTION: C
ROLL NO: 31
REGISTRATION: 180905218

SAMPLE QUESTION

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

int main(int argc, char *argv[])
{
        int rank,size,N,A[10],B[10],c,i;
        MPI_Init(&argc,&argv);
        MPI_Comm_rank(MPI_COMM_WORLD,&rank);
        MPI_Comm_size(MPI_COMM_WORLD,&size);

        if(rank==0)
        {
                N=size;
                fprintf(stdout,"Enter %d values: \n",N);
                fflush(stdout);
                for(i=0;i<N;i++)
                        scanf("%d",&A[i]);
        }
        MPI_Scatter(A,1,MPI_INT,&c,1,MPI_INT,0,MPI_COMM_WORLD);
        fprintf(stdout,"I have received %d in process %d\n",c,rank);
        fflush(stdout);

        c=c*c;
        MPI_Gather(&c,1,MPI_INT,B,1,MPI_INT,0,MPI_COMM_WORLD);

        if(rank==0)
        {
                fprintf(stdout,"The Result gathered in the root \n");
                fflush(stdout);
                for(i=0;i<N;i++)
                        fprintf(stdout,"%d\t",B[i]);
                fflush(stdout);
                fprintf(stdout,"\n");
        }

        MPI_Finalize();
        return 0;
}
```

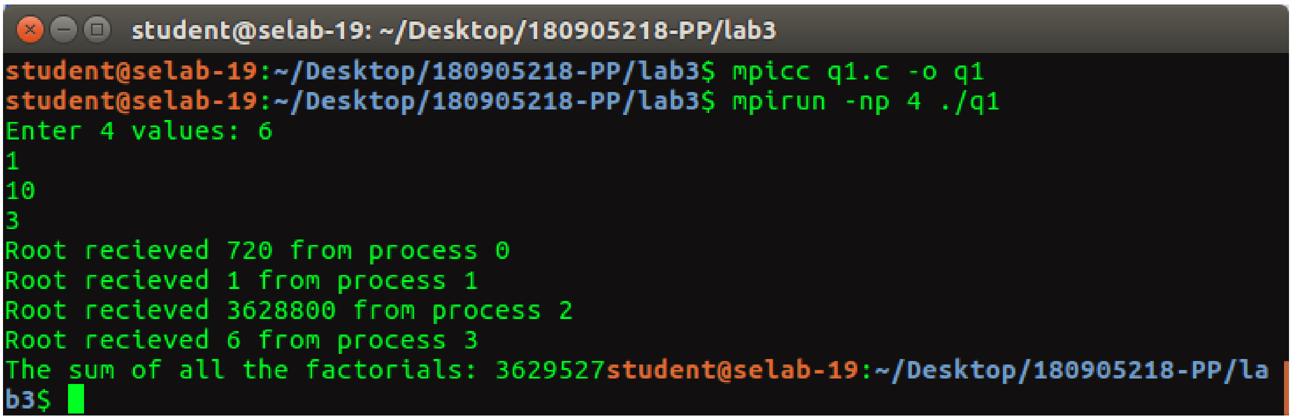


```
student@selab-19:~/Desktop/180905218-PP/lab3$ mpicc sample.c -o sample
student@selab-19:~/Desktop/180905218-PP/lab3$ mpirun -np 4 ./sample
Enter 4 values:
5
1
9
2
I have received 5 in process 0
The Result gathered in the root
25      1       81      4
I have received 1 in process 1
I have received 9 in process 2
I have received 2 in process 3
student@selab-19:~/Desktop/180905218-PP/lab3$ 
```

Q1: Factorial using n processes

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

int factorial(int x)
{
	int ans = 1;
	for(int i = x; i > 0; i--)
		ans = ans*i;
	return ans;
}

void main(int argc, char* argv[])
{
	int rank,size, c;
	int N;
	int sum=0;
	int a[N], b[N];

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

	if(rank == 0)
	{
		N=size;
		printf("Enter %d values: ",N);
		for(int i=0; i<N; i++)
		{
			scanf("%d", &a[i]);
		}
	}

	MPI_Scatter(a, 1, MPI_INT, &c, 1, MPI_INT, 0, MPI_COMM_WORLD);

	c = factorial(c);
```

```
    MPI_Gather(&c, 1, MPI_INT, b, 1, MPI_INT, 0, MPI_COMM_WORLD);

    if(rank == 0)
    {
        for(int i=0; i<N; i++)
        {
            printf("Root recieved %d from process %d \n", b[i], i);
            sum+=b[i];
        }
        printf("The sum of all the factorials: %d",sum);
    }

    MPI_Finalize();

}
```

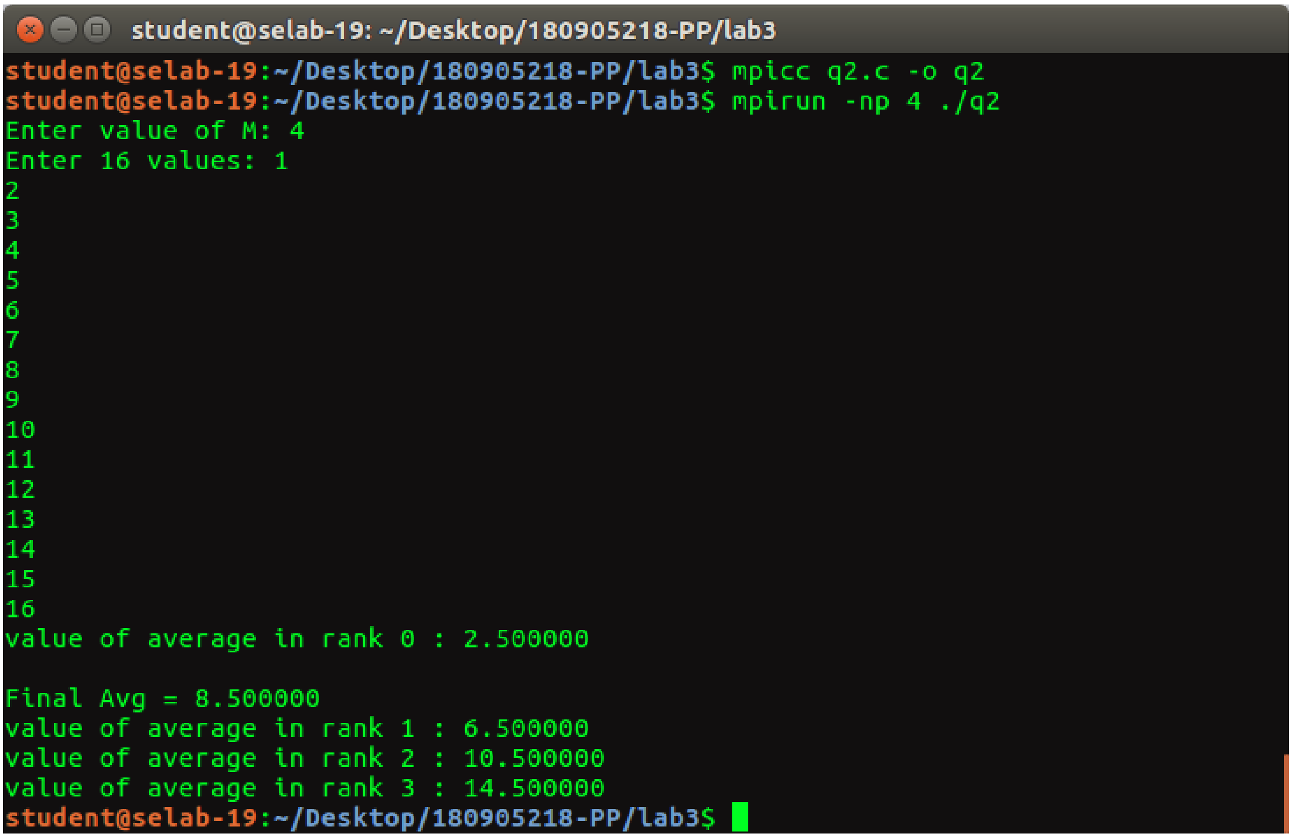


student@selab-19: ~/Desktop/180905218-PP/lab3

```
student@selab-19:~/Desktop/180905218-PP/lab3$ mpicc q1.c -o q1
student@selab-19:~/Desktop/180905218-PP/lab3$ mpirun -np 4 ./q1
Enter 4 values: 6
1
10
3
Root recieved 720 from process 0
Root recieved 1 from process 1
Root recieved 3628800 from process 2
Root recieved 6 from process 3
The sum of all the factorials: 3629527student@selab-19:~/Desktop/180905218-PP/la
b3$ 
```

Q2: Average of NxM array

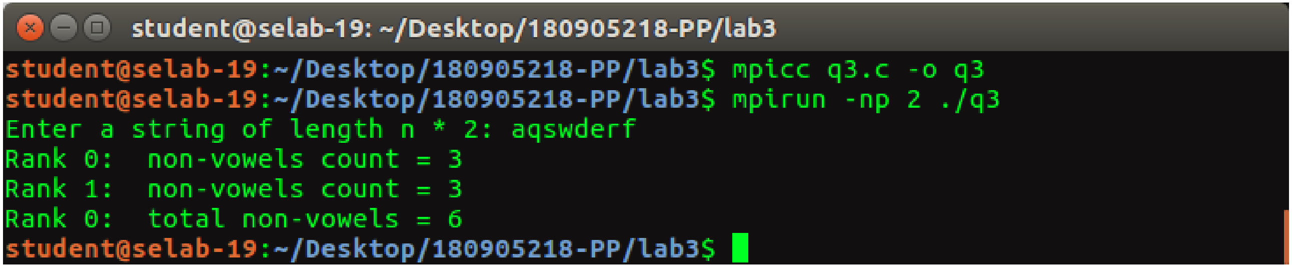
```
#include <stdio.h>
#include <mpi.h>


float avg(int *x, int M)
{
    float ans = 0.0;
    for(int i = 0; i < M; i++)
        ans = ans+x[i];
    return ans/M;
}

void main(int argc, char* argv[])
{
    int rank,size;
    float c;
    int N;
    int M;
    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&size);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);

    if(rank == 0)
    {
        N=size;
        printf("Enter value of M: ");
        scanf("%d", &M);
    }

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

    int b[M];
    float x[N];

    int a[N*M];

    if(rank == 0)
    {
        printf("Enter %d values: ", N*M);
        for(int i=0; i<N*M; i++)
        {
            scanf("%d", &a[i]);
        }

    }

    MPI_Scatter(a, M, MPI_INT, b, M, MPI_INT, 0, MPI_COMM_WORLD);

    c = avg(b, M);

    printf("value of average in rank %d : %f \n", rank, c);

    MPI_Gather(&c, 1, MPI_FLOAT, x, 1, MPI_FLOAT, 0, MPI_COMM_WORLD);

    if(rank == 0)
    {
        float ans = 0.0;
        for(int i = 0; i < N; i++)
            ans = ans+x[i];
        ans = ans/N;
        printf("\nFinal Avg = %f \n", ans);
    }

    MPI_Finalize();



}
```



```
student@selab-19:~/Desktop/180905218-PP/lab3$ mpicc q2.c -o q2
student@selab-19:~/Desktop/180905218-PP/lab3$ mpirun -np 4 ./q2
Enter value of M: 4
Enter 16 values: 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
value of average in rank 0 : 2.500000

Final Avg = 8.500000
value of average in rank 1 : 6.500000
value of average in rank 2 : 10.500000
value of average in rank 3 : 14.500000
student@selab-19:~/Desktop/180905218-PP/lab3$ 
```

Q3: Non vowels

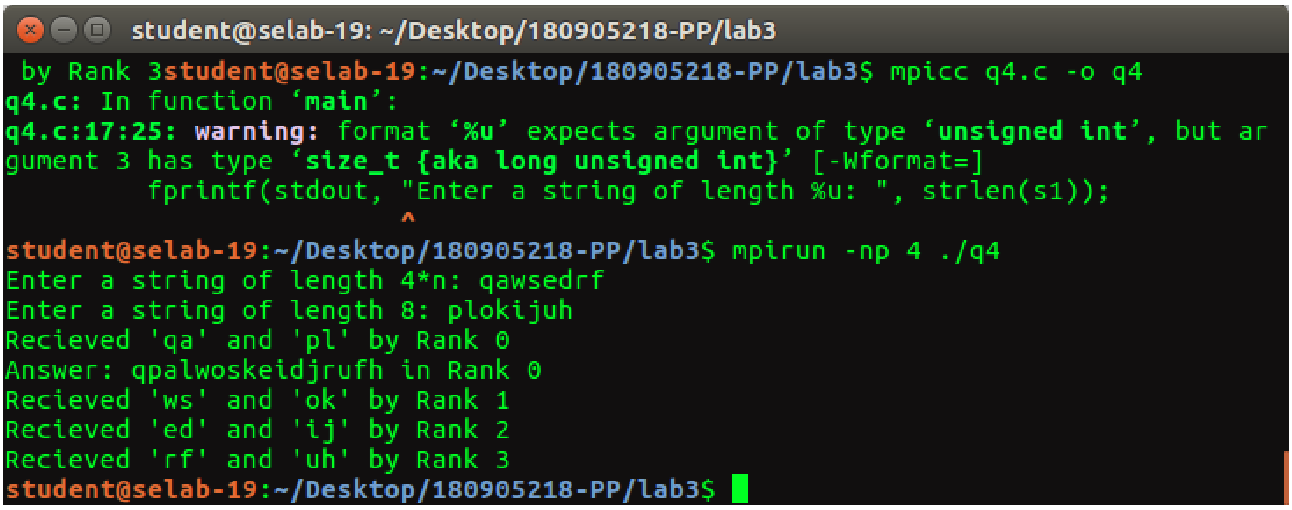
```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[]){
        int rank, size, len, l_v_c = 0, t_v_c = 0, *p_v_c;
    char str[100], recvStr[100];

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

    if(rank == 0){
        printf("Enter a string of length n * %d: ", size);
        scanf("%s", str);
        len = strlen(str);

        if(len%size != 0){
            printf("Invalid string.\n");
            MPI_Abort(MPI_COMM_WORLD, EXIT_FAILURE);
        }
    }
    MPI_Bcast(&len, 1, MPI_INT, 0, MPI_COMM_WORLD);

    MPI_Scatter(str, len/size, MPI_CHAR, recvStr, len/size, MPI_CHAR, 0,
MPI_COMM_WORLD);

    for(int i = 0; i < len/size; i++){
        char ch = recvStr[i];
        if(ch != 'a' && ch != 'e' && ch != 'i' && ch != 'o' && ch != 'u' &&
            ch != 'A' && ch != 'E' && ch != 'I' && ch != 'O' && ch != 'U') {
            l_v_c++;
        }
    }

    fprintf(stdout, "Rank %d:\t non-vowels count = %d\n", rank, l_v_c);
    fflush(stdout);

    p_v_c = (int*)malloc(size*sizeof(int));
    MPI_Gather(&l_v_c, 1, MPI_INT, p_v_c, 1, MPI_INT, 0, MPI_COMM_WORLD);

    if(rank == 0){
        for(int i = 0; i < size; i++){
            t_v_c += p_v_c[i];
        }
        fprintf(stdout, "Rank %d:\t total non-vowels = %d\n", rank, t_v_c);
        fflush(stdout);
    }
    MPI_Finalize();
    return 0;
}
```



```
student@selab-19:~/Desktop/180905218-PP/lab3$ mpicc q3.c -o q3
student@selab-19:~/Desktop/180905218-PP/lab3$ mpirun -np 2 ./q3
Enter a string of length n * 2: aqswderf
Rank 0:  non-vowels count = 3
Rank 1:  non-vowels count = 3
Rank 0:  total non-vowels = 6
student@selab-19:~/Desktop/180905218-PP/lab3$ 
```

Q4: Alternating string concat

```
#include <stdio.h>
#include "mpi.h"
#include <string.h>
int main(int argc, char *argv[])
{
    MPI_Init(&argc,&argv);
    int rank,size;
    MPI_Status status;
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);
    MPI_Comm_size(MPI_COMM_WORLD,&size);
    int m;
    char s1[100],c1[100],s2[100],c2[100],l[200],ans[200];
    if(rank==0){
        fprintf(stdout, "Enter a string of length %d*n: ", size);
        fflush(stdout);
        scanf("%s",s1);
        fprintf(stdout, "Enter a string of length %u: ", strlen(s1));
        fflush(stdout);
        scanf("%s",s2);
        m=strlen(s1)/size;
    }
    MPI_Bcast(&m,1,MPI_INT,0,MPI_COMM_WORLD);
    MPI_Scatter(s1,m,MPI_CHAR,c1,m,MPI_CHAR,0,MPI_COMM_WORLD);
    MPI_Scatter(s2,m,MPI_CHAR,c2,m,MPI_CHAR,0,MPI_COMM_WORLD);

    c1[m]='\0';
    c2[m]='\0';
      for(int i=0;i<2*m;i+=2){
        l[i]=c1[i/2];
        l[i+1]=c2[i/2];
    }
    l[2*m]='\0';
    fprintf(stdout, "Recieved '%s' and '%s' by Rank %d\n",c1,c2,rank);
    fflush(stdout);
    MPI_Gather(l,2*m,MPI_CHAR,ans,2*m,MPI_CHAR,0,MPI_COMM_WORLD);
    if(rank==0){
        fprintf(stdout, "Answer: %s in Rank %d\n",ans,rank);
        fflush(stdout);
    }
    MPI_Finalize();
    return 0;
}
```



```
student@selab-19: ~/Desktop/180905218-PP/lab3
 by Rank 3student@selab-19:~/Desktop/180905218-PP/lab3$ mpicc q4.c -o q4
q4.c: In function ‘main’:
q4.c:17:25: warning: format ‘%u’ expects argument of type ‘unsigned int’, but ar
gument 3 has type ‘size_t {aka long unsigned int}’ [-Wformat=]
         fprintf(stdout, "Enter a string of length %u: ", strlen(s1));
                         ^
student@selab-19:~/Desktop/180905218-PP/lab3$ mpirun -np 4 ./q4
Enter a string of length 4*n: qawsedrf
Enter a string of length 8: plokijuh
Recieved 'qa' and 'pl' by Rank 0
Answer: qpalwoskeidjrufh in Rank 0
Recieved 'ws' and 'ok' by Rank 1
Recieved 'ed' and 'ij' by Rank 2
Recieved 'rf' and 'uh' by Rank 3
student@selab-19:~/Desktop/180905218-PP/lab3$ 
```