

# PP LAB-4

1.

```
#include <mpi.h>
#include <stdio.h>
#include <string.h>
int main (int argc, char *argv[])
{
    int rank, size;
    int i = 0, j;
    int k = 0, fac=1, ans[1000], sum=0;
    int n, a[100][100], b[100];
    float x, y, area, pi1;
    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    // Set the error handler to MPI_ERRORS_RETURN
    MPI_Errhandler_set(MPI_COMM_WORLD, MPI_ERRORS_RETURN);

    MPI_Comm_size(MPI_COMM_WORLD, &size);

    int error = MPI_Bcast(&size, 1, MPI_INT, 0, MPI_COMM_WORLD);
    if (error != MPI_SUCCESS)
    {
        char s[100];
        int len, class1;
        MPI_Error_string(error, s, &len);
        MPI_Error_class(error, &class1);
        fprintf(stderr, "Error description is %s", s);
        fflush(stderr);
        fprintf(stderr, "Error class is %d", class1);
        fflush(stderr);
    }

    x = (float)(rank+1)/size;
    y = 4.f/(1+x*x);
    area = (1/(float)size)*y;
    MPI_Reduce(&area, &pi1, 1, MPI_FLOAT, MPI_SUM, 0,
    MPI_COMM_WORLD);
    if (rank == 0)
    {
```

```

fprintf(stdout, "%f\n", pi1);
fflush(stdout);
}

```

```

MPI_Finalize();
return 0;
}

```

```

Student@dblab-hp-12:~/Documents/week4$ mpicc q1.c -o q1.o
Student@dblab-hp-12:~/Documents/week4$ mpirun -np 4 ./q1.o
33

```

2.

```

#include <mpi.h>
#include <stdio.h>
#include <string.h>
int main (int argc, char *argv[])
{
    int rank, size;
    int i = 0, j;
    int k = 0, fac=1, ans[1000], sum=0;
    int n, a[100][100], b[100];
    float x, y, area, pi1;
    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    // Set the error handler to MPI_ERRORS_RETURN
    MPI_Errhandler_set(MPI_COMM_WORLD, MPI_ERRORS_RETURN);

    MPI_Comm_size(MPI_COMM_WORLD, &size);

    int error = MPI_Bcast(&size, 1, MPI_INT, 0, MPI_COMM_WORLD);
    if (error != MPI_SUCCESS)
    {
        char s[100];
        int len, class1;
        MPI_Error_string(error, s, &len);
        MPI_Error_class(error, &class1);
        fprintf(stderr, "Error description is %s", s);
        fflush(stderr);
        fprintf(stderr, "Error class is %d", class1);
        fflush(stderr);
    }
}

```

```

}

x = (float)(rank+1)/size;
y = 4.f/(1+x*x);
area = (1/(float)size)*y;
MPI_Reduce(&area, &pi1, 1, MPI_FLOAT, MPI_SUM, 0, MPI_COMM_WORLD);
if (rank == 0)
{
    fprintf(stdout, "%f\n", pi1);
    fflush(stdout);
}

MPI_Finalize();
return 0;
}

```

```

Student@dblab-hp-12:~/Documents/week4$ mpicc q2.c -o q2.o
Student@dblab-hp-12:~/Documents/week4$ mpirun -np 4 ./q2.o
3.146801

```

3.

```

#include <stdio.h>
#include <mpi.h>
void ErrorHandler(int error_code)
{
    char error_string[MPI_MAX_ERROR_STRING];
    int length_of_error_string,error_class;
    MPI_Error_class(error_code,&error_class);
    MPI_Error_string(error_code,error_string,&length_of_error_string);
    if(error_code!=0)
        printf("error class %d \n error string %s\n",error_class,error_string );
}
void main (int a,char *b[])
{
    int rank, ele,size;
    int a1[3][3];
    int b1[3];
    int count=0;
    int tc = 0;
    int error_code;
    MPI_Init(&a, &b);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Errhandler_set(MPI_COMM_WORLD,MPI_ERRORS_RETURN);

```

```

error_code=MPI_Comm_size(MPI_COMM_WORLD,&size);
ErrorHandler(error_code);

if (rank == 0)
{
printf("Enter elements into matrix\n");
for (int i = 0; i < 3; i++)
{
for (int j = 0; j < 3; j++)
{
scanf("%d", &a1[i][j]);
}
}

printf("Enter element to count \n");
scanf("%d", &ele);

}

MPI_Bcast(&ele, 1, MPI_INT, 0, MPI_COMM_WORLD);
MPI_Scatter(a1 , 3, MPI_INT, b1, 3, MPI_INT, 0, MPI_COMM_WORLD);
for (int i = 0; i < 3; i++)
{
printf("%d ",b1[i]);
if (b1[i] == ele)
{
count ++;
}
}

printf("\nProcess %d found %d occurrences\n",rank,count);
MPI_Reduce(&count, &tc, 1, MPI_INT, MPI_SUM, 0, MPI_COMM_WORLD);
if (rank == 0)
{
printf("\nNumber of occurrences is %d\n",tc);
}
MPI_Finalize();
}

```

```

Student@dblab-hp-12:~/Documents/week4$ mpirun -np 3 ./q3.o
Enter elements into matrix
3 4 2
1 2 1
9 6 9
Enter element to count
9
3 4 2
Process 0 found 0 occurrences
1 2 9 6 9
Process 2 found 2 occurrences
1
Process 1 found 0 occurrences
Number of occurrences is 2

```

4.

```

#include "mpi.h"
#include <stdio.h>
#include <string.h>
void ErrorHandler(int error_code){
if (error_code != MPI_SUCCESS){
char error_string[BUFSIZ];
int length_of_error_string, error_class;
MPI_Error_class(error_code, &error_class);
MPI_Error_string(error_code, error_string, &length_of_error_string);
printf("%d %s\n", error_class, error_string);
}
}
int main (int argc, char* argv[]) {
int rank, size, error_code;
int i = 0, j;
int k = 0, fac = 1, ans[1000], sum = 0;
int n, a[100][100], b[100];
MPI_Init(&argc, &argv);
error_code = MPI_Comm_rank(MPI_COMM_WORLD, &rank);
error_code = MPI_Comm_size(MPI_COMM_WORLD, &size);
if (rank == 0)
{
printf("Enter the elements of i/p matrix \n");
for (i = 0; i < 4; i++) {
for (j = 0; j < 4; j++) {scanf("%d", &a[i][j]);
}
}
printf("\n");
}
}

```

```
error_code = MPI_Scatter(a, 100, MPI_INT, b, 100, MPI_INT, 0,
MPI_COMM_WORLD);
error_code = MPI_Scan(b, ans, 4, MPI_INT, MPI_SUM, MPI_COMM_WORLD);
ErrorHandler(error_code);
printf("\n");
for (i = 0; i < 4; i++) {
printf("%d ", ans[i]);
}
printf("\n");
MPI_Finalize();
return 0;
}
```

```
Student@dblab-hp-12:~/Documents/week4$ mpicc q4.c -o q4.out
Student@dblab-hp-12:~/Documents/week4$ mpirun -np 4 ./q4.out
Enter the elements of i/p matrix
1 2 3 4
1 2 3 1
1 1 1 1
2 1 2 1

1 2 3 4
2 4 6 5
3 5 7 6
5 6 9 7
```