SAMPLE PROGRAM

```c
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>
float *create_rand_nums(int  n)
{
   float *rnd=(float *) malloc(sizeof(float)*n);
   int i;
   for(i=0;i<n;i++)
   {
                rnd[i]=rand();
   }
        return rnd;
}


int main(int argc, char *argv[])
{
   MPI_Init(&argc,&argv);
   int id;
        int p,i;
        int num_elements_per_proc=1;
        MPI_Comm_rank(MPI_COMM_WORLD, &id);
   MPI_Comm_size(MPI_COMM_WORLD, &p);
   float *rand_nums = NULL;
   if(id==0)
        {
           rand_nums = create_rand_nums(p);
        }
```

```c
        float *sub_rand_nums = (float *)malloc(sizeof(float) * p);

    MPI_Scatter(rand_nums, num_elements_per_proc, MPI_FLOAT, sub_rand_nums, 1, MPI_FLOAT,
0, MPI_COMM_WORLD);

        if(id>0)

        {

                printf("\nData recieved: ");

        for( i=0;i<p;i++)

        {

                printf("%.2f ",sub_rand_nums[i]);

                }

                printf("\n");

        }

    MPI_Finalize();

    return 0;

}
```
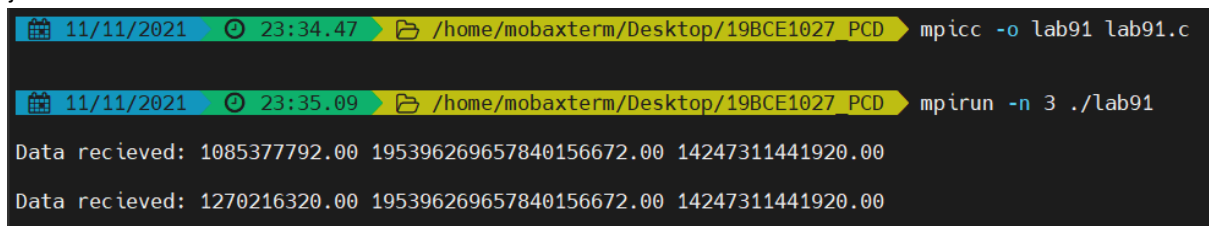
```
11/11/2021    23:34.47    /home/mobaxterm/Desktop/19BCE1027_PCD    mpicc -o lab91 lab91.c

11/11/2021    23:35.09    /home/mobaxterm/Desktop/19BCE1027_PCD    mpirun -n 3 ./lab91

Data recieved: 1085377792.00 195396269657840156672.00 14247311441920.00

Data recieved: 1270216320.00 195396269657840156672.00 14247311441920.00
```

Create a random number and broadcast to each other by using Gather Function.

```c
#include "mpi.h"

#include <stdio.h>

#include <stdlib.h>

float *create_rand_nums(int  n)

{

    float *rnd=(float *) malloc(sizeof(float)*n);

    int i;

    for(i=0;i<n;i++)

    {

                rnd[i]=rand();

    }

        return rnd;
```

```c
}

int main(int argc, char *argv[])
{
    MPI_Init(&argc,&argv);
    int id;
        int p,i;
        int num_elements_per_proc=1;
        MPI_Comm_rank(MPI_COMM_WORLD, &id);
    MPI_Comm_size(MPI_COMM_WORLD, &p);
    float *rand_nums = NULL;
    if(id==0)
        {
            rand_nums = create_rand_nums(p);
        }
        float *sub_rand_nums = (float *)malloc(sizeof(float) * p);
         MPI_Gather(&rand_nums, 1, MPI_FLOAT, sub_rand_nums, 1, MPI_FLOAT, 0,
MPI_COMM_WORLD);
        if(id>0)
        {
                printf("\nData recieved: ");
        for( i=0;i<p;i++)
        {
                printf("%.2f ",sub_rand_nums[i]);
                }
                printf("\n");
        }
    MPI_Finalize();
    return 0;
}
```

```
📅 11/11/2021  🕐 23:59.42  📂 /home/mobaxterm/Desktop/19BCE1027_PCD  mpicc -o lab92 lab92.c

📅 11/11/2021  🕐 23:59.56  📂 /home/mobaxterm/Desktop/19BCE1027_PCD  mpirun -n 3 ./lab92
Data recieved: 0.00 0.00 0.00
Data recieved: 0.00 -nan 0.00
```