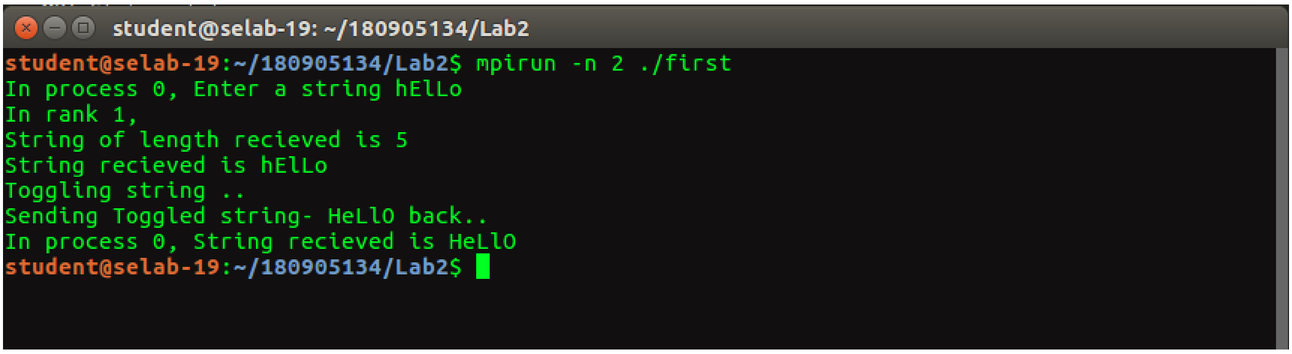
SANSKAAR PATNI
180905134 CSE C-23
PCAP LAB 2

1.CODE

```
#include "mpi.h"
#include "mpi.h"
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int main(int argc,char *argv[]){
	int rank,size;
	int length;
	char str[10];
	MPI_Init(&argc,&argv);
	MPI_Comm_rank(MPI_COMM_WORLD,&rank);
	MPI_Comm_size(MPI_COMM_WORLD,&size);
	MPI_Status status;
	if(rank==0){
		fprintf(stdout,"In process %d, Enter a string ",rank);
		fflush(stdout);
		scanf("%s",str);
		length=strlen(str);
		MPI_Ssend(&length,1,MPI_INT,1,1,MPI_COMM_WORLD);
		MPI_Ssend(str,length,MPI_CHAR,1,2,MPI_COMM_WORLD);
		MPI_Recv(str,length,MPI_CHAR,1,3,MPI_COMM_WORLD,&status);
		fprintf(stdout,"In process %d, String recieved is %s\n",rank,str);
		fflush(stdout);
	}
	else{
		MPI_Recv(&length,1,MPI_INT,0,1,MPI_COMM_WORLD,&status);
		fprintf(stdout,"In rank %d,\nString of length recieved is %d\n",rank,length);
		fflush(stdout);
		MPI_Recv(str,length,MPI_CHAR,0,2,MPI_COMM_WORLD,&status);
		fprintf(stdout,"String recieved is %s\n",str);
		fflush(stdout);
		fprintf(stdout,"Toggling string ..\n");
		fflush(stdout);
		for(int i=0;i<length;i++){
			if(islower(str[i])>0){
				str[i]=toupper(str[i]);
			}
			else{
				str[i]=tolower(str[i]);
			}
		}
		fprintf(stdout,"Sending Toggled string- %s back..\n",str);
		fflush(stdout);
		MPI_Ssend(str,length,MPI_CHAR,0,3,MPI_COMM_WORLD);
	}
	MPI_Finalize();
	return 0;
```

}

SCREENSHOT

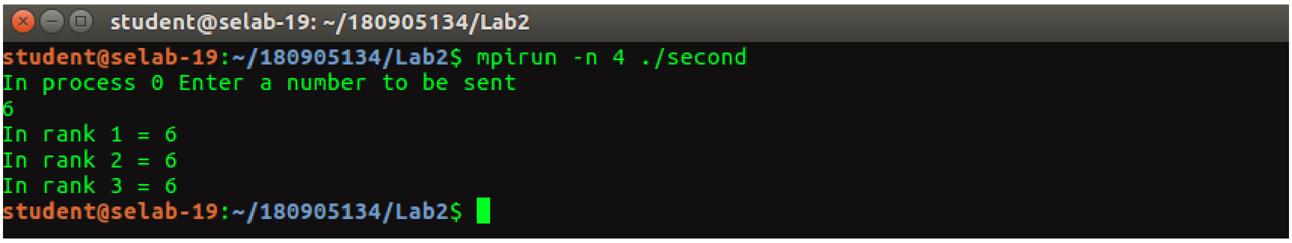


```
student@selab-19:~/180905134/Lab2$ mpirun -n 2 ./first
In process 0, Enter a string hElLo
In rank 1,
String of length recieved is 5
String recieved is hElLo
Toggling string ..
Sending Toggled string- HeLlO back..
In process 0, String recieved is HeLlO
student@selab-19:~/180905134/Lab2$ 
```

2.CODE

```
#include "mpi.h"
#include <stdio.h>
#include <math.h>
int main(int argc,char *argv[])
{
        int rank,size;
        int x,a;
        MPI_Init(&argc,&argv);
        MPI_Comm_rank(MPI_COMM_WORLD,&rank);
        MPI_Comm_size(MPI_COMM_WORLD,&size);
        MPI_Status status;
        if(rank==0)
        {
                fprintf(stdout,"In process %d\nEnter a number to be sent \n",rank);
                fflush(stdout);
                scanf("%d",&a);
                for(int i=1;i<size;i++)
                {
                        MPI_Send(&a,1,MPI_INT,i,i,MPI_COMM_WORLD);
                }
        }
        else
        {
        MPI_Recv(&a,1,MPI_INT,0,rank,MPI_COMM_WORLD,&status);
        fprintf(stdout,"In rank %d = %d\n",rank,a);
        fflush(stdout);
        }
        MPI_Finalize();
        return 0;
}
```

SCREENSHOT:

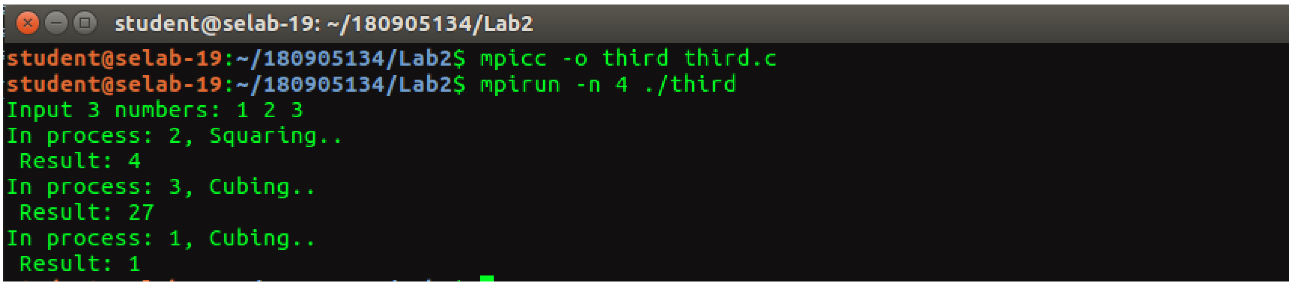


```
student@selab-19:~/180905134/Lab2$ mpirun -n 4 ./second
In process 0 Enter a number to be sent
6
In rank 1 = 6
In rank 2 = 6
In rank 3 = 6
student@selab-19:~/180905134/Lab2$
```

3.CODE

```
#include<stdio.h>
#include "mpi.h"
#include<stdlib.h>
int main(int argc, char* argv[])
{
        int rank;
        int val = 0;
        int size;
        int arr[25];
        MPI_Status status;
        int b[25];
        MPI_Init(&argc, &argv);
        MPI_Buffer_attach(b, 25 * sizeof(int));
        MPI_Comm_rank(MPI_COMM_WORLD, &rank);
        MPI_Comm_size(MPI_COMM_WORLD, &size);
        if (rank == 0)
        {
                fprintf(stdout, "Input %d numbers: ", size-1);
                fflush(stdout);
                for (int i = 1;i < size;i++)
                        scanf("%d", &arr[i-1]);
                for (int i = 1;i < size;i++)
                {
                        MPI_Bsend(&arr[i - 1], 1, MPI_INT, i, i, MPI_COMM_WORLD);
                }
        }
        else
        {
                MPI_Recv(&val, 1, MPI_INT, 0, rank, MPI_COMM_WORLD, &status);
                if (rank % 2 == 0)
                {
                        fprintf(stdout,"In process: %d, Squaring..\n Result: %d\n",rank,(val*val));
                        fflush(stdout);
                }
                else
                {
                        fprintf(stdout,"In process: %d, Cubing..\n Result: %d\n",rank,(val*val*val));
                        fflush(stdout);
                }
        }
        MPI_Finalize();
        return 0;
}
```

SCREENSHOT

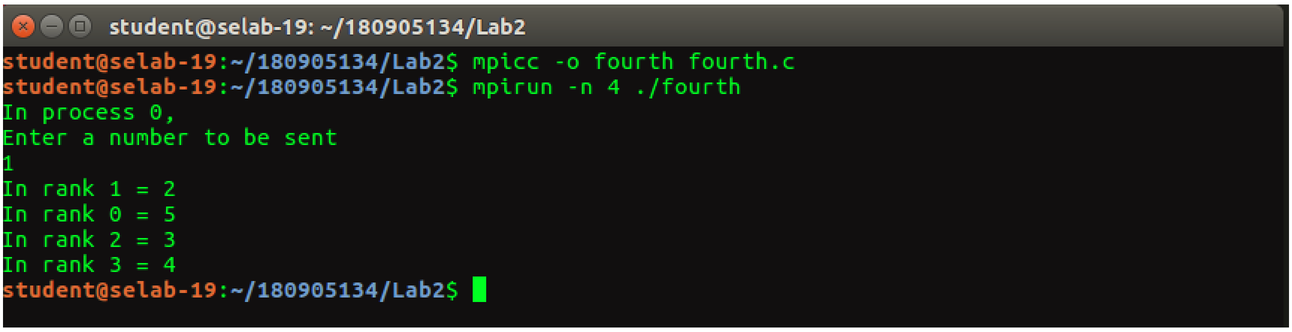


```
student@selab-19:~/180905134/Lab2$ mpicc -o third third.c
student@selab-19:~/180905134/Lab2$ mpirun -n 4 ./third
Input 3 numbers: 1 2 3
In process: 2, Squaring..
 Result: 4
In process: 3, Cubing..
 Result: 27
In process: 1, Cubing..
 Result: 1
```

4.CODE

```
#include "mpi.h"
#include <stdio.h>
int main(int argc,char *argv[])
{
	int rank,size;
	int x,a;
	MPI_Init(&argc,&argv);
	MPI_Comm_rank(MPI_COMM_WORLD,&rank);
	MPI_Comm_size(MPI_COMM_WORLD,&size);
	MPI_Status status;
	if(rank==0)
	{
		fprintf(stdout,"In process %d,\nEnter a number to be sent \n",rank);
		fflush(stdout);
		scanf("%d",&a);
		a=a+1;
		MPI_Send(&a,1,MPI_INT,rank+1,1,MPI_COMM_WORLD);
		MPI_Recv(&a,1,MPI_INT,size-1,1,MPI_COMM_WORLD,&status);
		fprintf(stdout,"In rank %d = %d\n",rank,a);
		fflush(stdout);
	}
	else
	{
		MPI_Recv(&a,1,MPI_INT,rank-1,1,MPI_COMM_WORLD,&status);
		fprintf(stdout,"In rank %d = %d\n",rank,a);
		fflush(stdout);
		a=a+1;
		MPI_Send(&a,1,MPI_INT,(rank+1)%size,1,MPI_COMM_WORLD);
	}
	MPI_Finalize();
	return 0;
}
```

SCREENSHOT:



```
student@selab-19:~/180905134/Lab2$ mpicc -o fourth fourth.c
student@selab-19:~/180905134/Lab2$ mpirun -n 4 ./fourth
In process 0,
Enter a number to be sent
1
In rank 1 = 2
In rank 0 = 5
In rank 2 = 3
In rank 3 = 4
student@selab-19:~/180905134/Lab2$
```