Name: Ankita Ghosh Sec: A Rollno: 41

Regno: 180905354 Sem: VI Branch: CSE

PARALLEL PROGRAMMING LAB 2

**Q1**.

Program:

#include <stdio.h>

#include "mpi.h"

#include <stdlib.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

int main(int argc, char \*argv[])

{

int rank, size;

char str[20]; char nstr[20];

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 string: ");

scanf("%s",str);

int len= strlen(str);

MPI\_Ssend(&len,1,MPI\_INT,1,1,MPI\_COMM\_WORLD);

MPI\_Ssend(str,len+1,MPI\_CHAR,1,1,MPI\_COMM\_WORLD);

printf("string sent from process 0 to 1\n");

MPI\_Recv(nstr,len+1,MPI\_CHAR,1,1, MPI\_COMM\_WORLD, &status);

printf("string recieved in process 1: ");

puts(nstr);

}

else if(rank==1)

{

char str[20]; int len;

MPI\_Recv(&len,1,MPI\_INT,0,1, MPI\_COMM\_WORLD, &status);

MPI\_Recv(str,len+1,MPI\_CHAR,0,1, MPI\_COMM\_WORLD, &status);

printf("string recieved from process 0 to 1\n");

for(int i=0;i<len;i++)

{

if(str[i]>='A' && str[i]<='Z')

str[i] = str[i]+32;

else if(str[i]>='a' && str[i]<='z')

str[i] = str[i]-32;

}

MPI\_Ssend(str,len+1,MPI\_CHAR,0,1,MPI\_COMM\_WORLD);

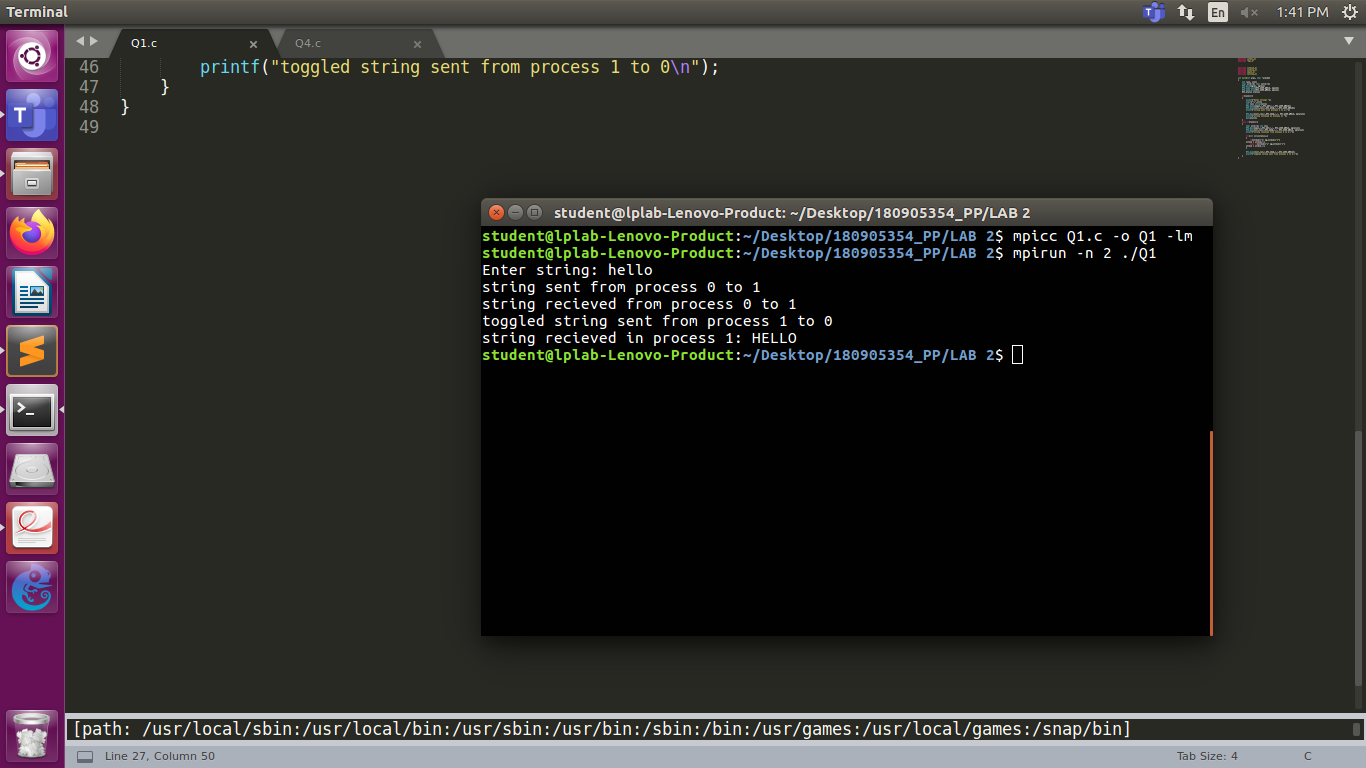
printf("toggled string sent from process 1 to 0\n");

}

MPI\_Finalize();

}

Output:



**Q2**.

Program:

#include <stdio.h>

#include "mpi.h"

#include <stdlib.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

int main(int argc, char \*argv[])

{

int rank, size; int num;

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 number: ");

scanf("%d",&num);

for(int i=1;i<size;i++)

MPI\_Send(&num,1,MPI\_INT,i,1,MPI\_COMM\_WORLD);

}

else

{

int num;

MPI\_Recv(&num,1,MPI\_INT,0,1, MPI\_COMM\_WORLD, &status);

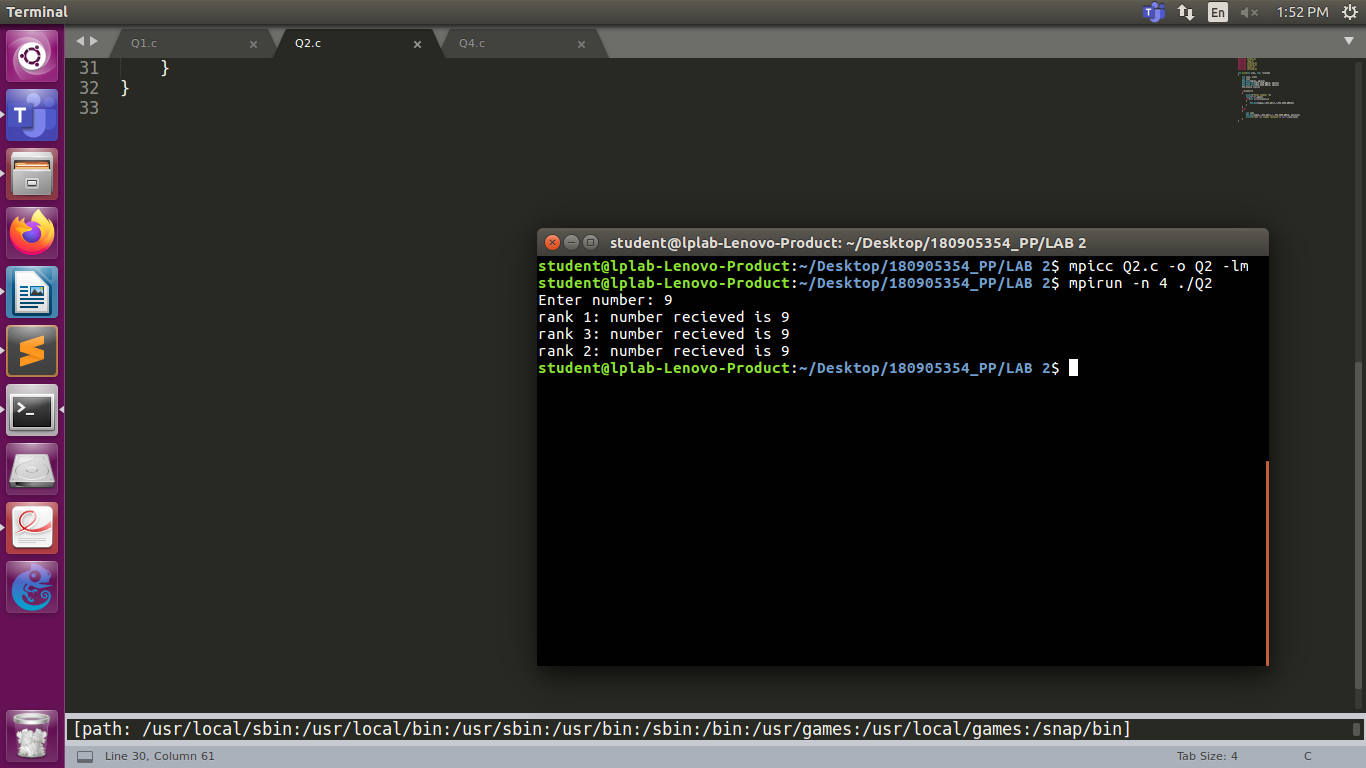
printf("rank %d: number recieved is %d\n",rank,num);

}

MPI\_Finalize();

}

Output:



**Q3**.

Program:

#include <stdio.h>

#include "mpi.h"

#include <stdlib.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

int main(int argc, char \*argv[])

{

int rank, size;

int num;

MPI\_Init(&argc, &argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

MPI\_Status status;

if(rank==0)

{

char buffer[500]; int bsize=500;

MPI\_Buffer\_attach(buffer, bsize);

int arr[size];

printf("Enter array elements: ");

for(int i=0;i<size;i++)

scanf("%d",&arr[i]);

printf("rank %d: number=%d value=%d\n",rank,arr[0],arr[0]\*arr[0]);

for(int i=1;i<size;i++)

{

MPI\_Bsend(&arr[i],1,MPI\_INT,i,1,MPI\_COMM\_WORLD);

}

MPI\_Buffer\_detach(&buffer, &bsize);

}

else if(rank%2==0)

{

int num;

MPI\_Recv(&num,1,MPI\_INT,0,1, MPI\_COMM\_WORLD, &status);

printf("rank %d: number=%d value=%d\n",rank,num,num\*num);

}

else

{

int num;

MPI\_Recv(&num,1,MPI\_INT,0,1, MPI\_COMM\_WORLD, &status);

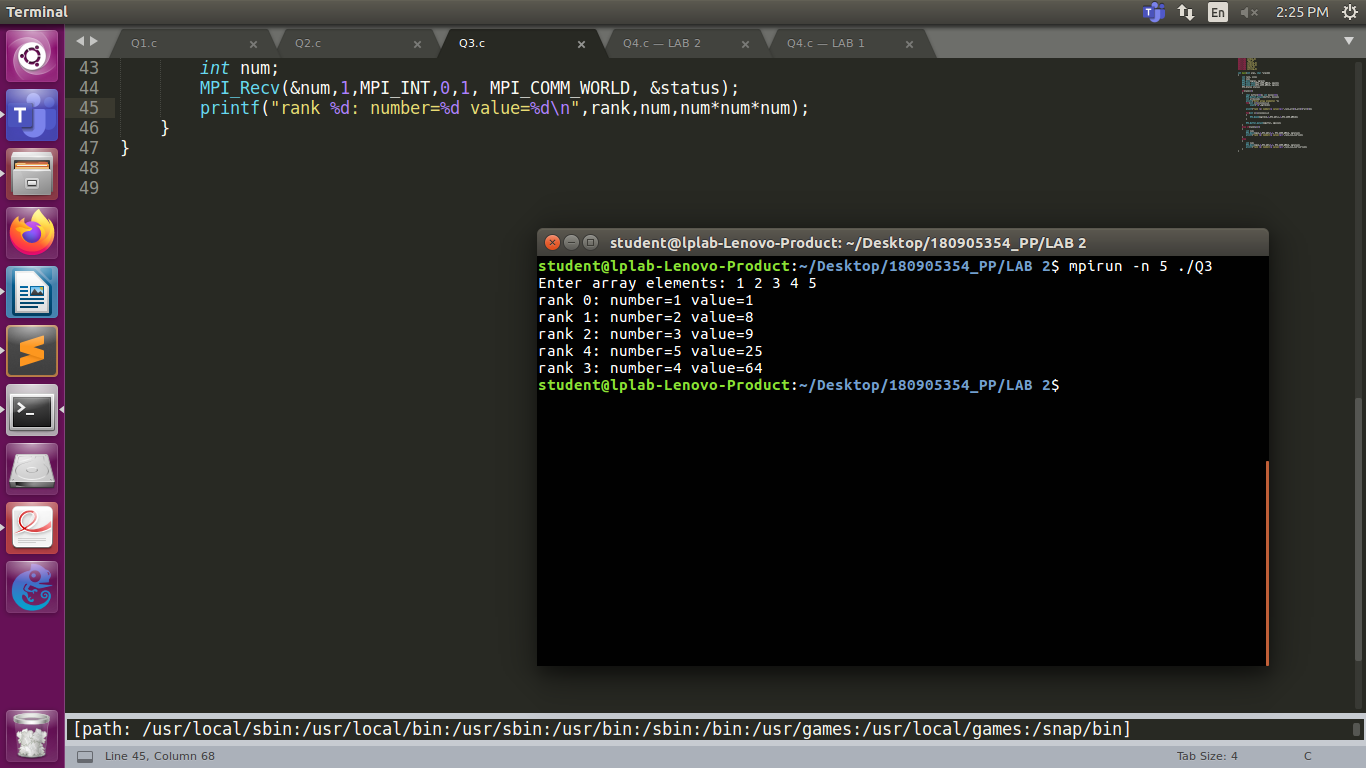
printf("rank %d: number=%d value=%d\n",rank,num,num\*num\*num);

}

MPI\_Finalize();

}

Output:



**Q4**.

Program:

#include <stdio.h>

#include "mpi.h"

#include <stdlib.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

int main(int argc, char \*argv[])

{

int rank, size;

int num;

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 number: ");

scanf("%d",&num);

MPI\_Send(&num,1,MPI\_INT,1,1,MPI\_COMM\_WORLD);

MPI\_Recv(&num,1,MPI\_INT,size-1,1, MPI\_COMM\_WORLD, &status);

printf("rank %d: number=%d\n",rank,num);

}

else

{

int num;

MPI\_Recv(&num,1,MPI\_INT,rank-1,1, MPI\_COMM\_WORLD, &status);

printf("rank %d: number=%d\n",rank,num);

num=num+1;

MPI\_Send(&num,1,MPI\_INT,(rank+1)%size,1,MPI\_COMM\_WORLD);

}

MPI\_Finalize();

}

Output:

