ANKITA GHOSH

180905354

Sec: A Rollno: 41

PARALLEL PROGRAMMING LAB 3

**Q1.**

Program:  
#include "mpi.h"

#include <stdio.h>

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

{

int rank,size,N,A[10],B[10],c,fact,i;

MPI\_Init(&argc, &argv);

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

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

MPI\_Status status;

if(rank==0)

{

N=size;

printf("Enter %d values:\n",N);

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

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

}

MPI\_Scatter(A,1,MPI\_INT,&c,1,MPI\_INT,0,MPI\_COMM\_WORLD);

fact=1;

for(i=1;i<=c;i++)

{

fact=fact\*i;

}

printf("Rank %d: number=%d factorial=%d \n",rank,c,fact);

MPI\_Gather(&fact,1,MPI\_INT,B,1,MPI\_INT,0,MPI\_COMM\_WORLD);

if(rank==0)

{

int sum=0;

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

sum=sum+B[i];

printf("Sum of the factorials: %d\n",sum);

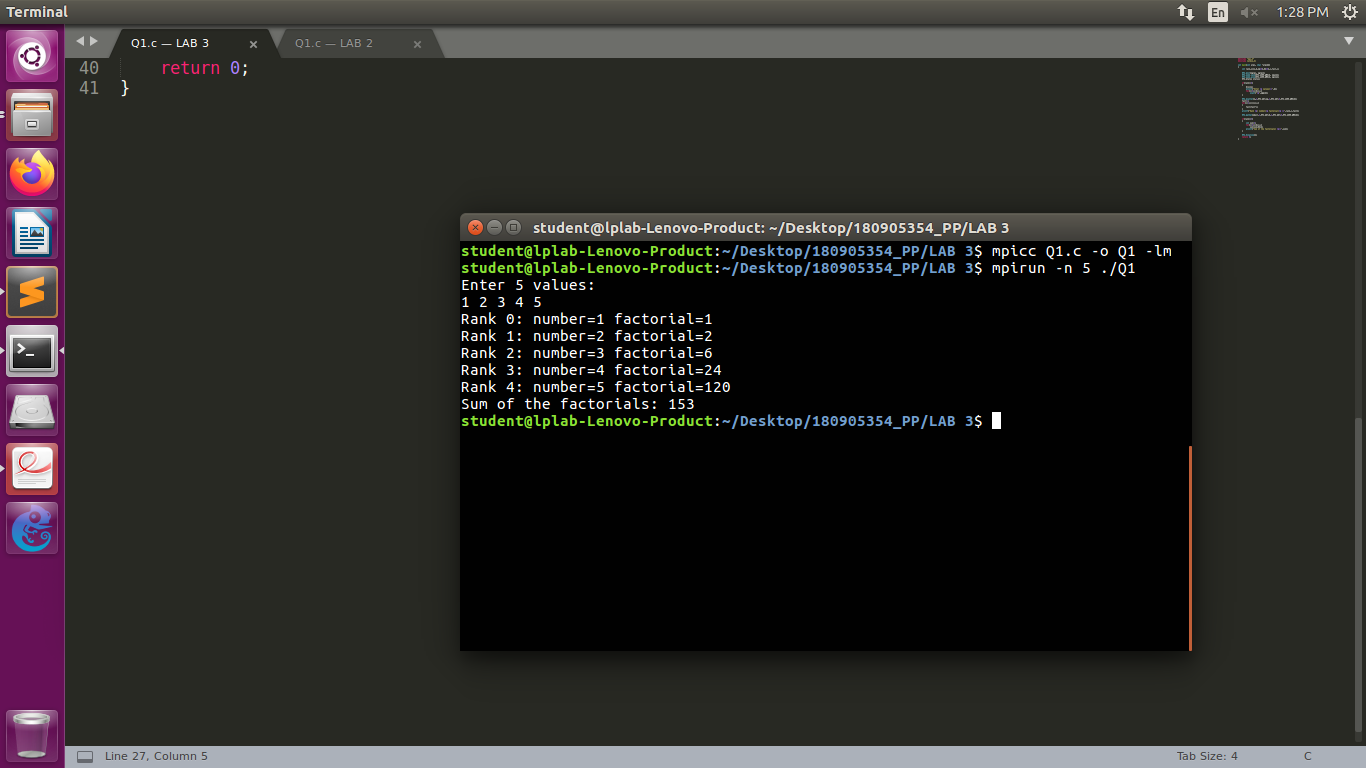
}

MPI\_Finalize();

return 0;

}

Output:



**Q2.**

Program:  
#include "mpi.h"

#include <stdio.h>

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

{

int rank,size,N,M,i;

int A[256],buff[256]; float B[256],sum,avg;

MPI\_Init(&argc, &argv);

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

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

MPI\_Status status;

if(rank==0)

{

N=size;

printf("Enter the value of M: ");

scanf("%d",&M);

printf("Enter %d values:\n",N\*M);

for(i=0;i<N\*M;i++)

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

}

MPI\_Bcast(&M,1,MPI\_INT,0,MPI\_COMM\_WORLD);

MPI\_Scatter(A,M,MPI\_INT,buff,M,MPI\_INT,0,MPI\_COMM\_WORLD);

sum=0; avg=0;

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

{

sum=sum+buff[i];

}

avg=sum/M;

printf("Rank %d: average=%lf \n",rank,avg);

MPI\_Gather(&avg,1,MPI\_FLOAT,B,1,MPI\_FLOAT,0,MPI\_COMM\_WORLD);

if(rank==0)

{

float tsum=0; float tavg=0;

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

tsum=tsum+B[i];

tavg=tsum/N;

printf("Total average: %lf\n",tavg);

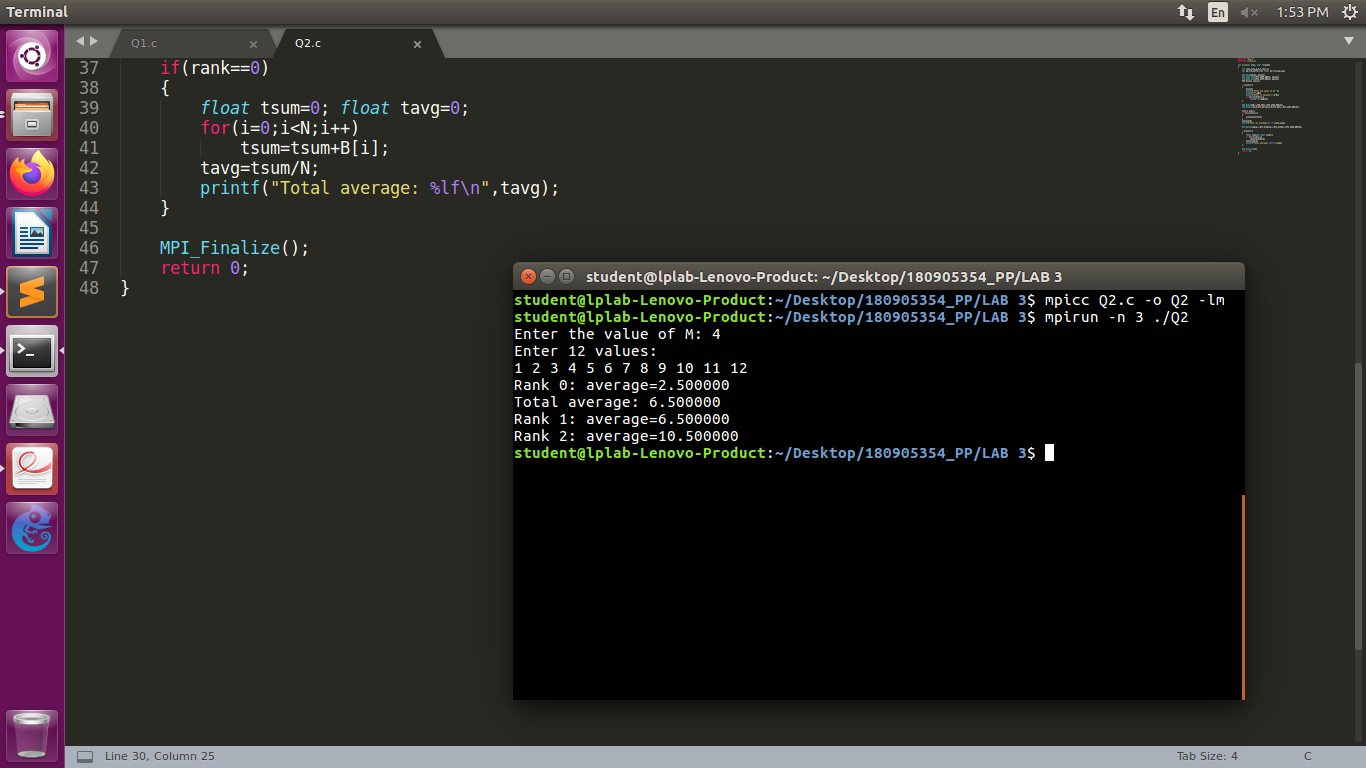
}

MPI\_Finalize();

return 0;

}

Output:



**Q3.**

Program:  
#include "mpi.h"

#include <stdio.h>

#include <string.h>

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

{

int rank,size,N,i,vcount,count,rlen;

char str[100],buff[100]; int B[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)

{

N=size;

printf("Enter the string: ");

scanf("%s",str);

rlen=strlen(str)/N;

}

MPI\_Bcast(&rlen,1,MPI\_INT,0,MPI\_COMM\_WORLD);

MPI\_Scatter(str,rlen,MPI\_CHAR,buff,rlen,MPI\_CHAR,0,MPI\_COMM\_WORLD);

vcount=0; count=0;

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

{

if(buff[i]=='a' || buff[i]=='e' || buff[i]=='i' || buff[i]=='o' || buff[i]=='u' || buff[i]=='A' || buff[i]=='E' || buff[i]=='I' || buff[i]=='O' || buff[i]=='U')

vcount++;

}

count=rlen-vcount;

MPI\_Gather(&count,1,MPI\_INT,B,1,MPI\_INT,0,MPI\_COMM\_WORLD);

int tcount=0;

if(rank==0)

{

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

{

tcount=tcount+B[i];

printf("Rank:%d Count=%d\n",i,B[i]);

}

printf("Total count of non-vowels: %d\n",tcount);

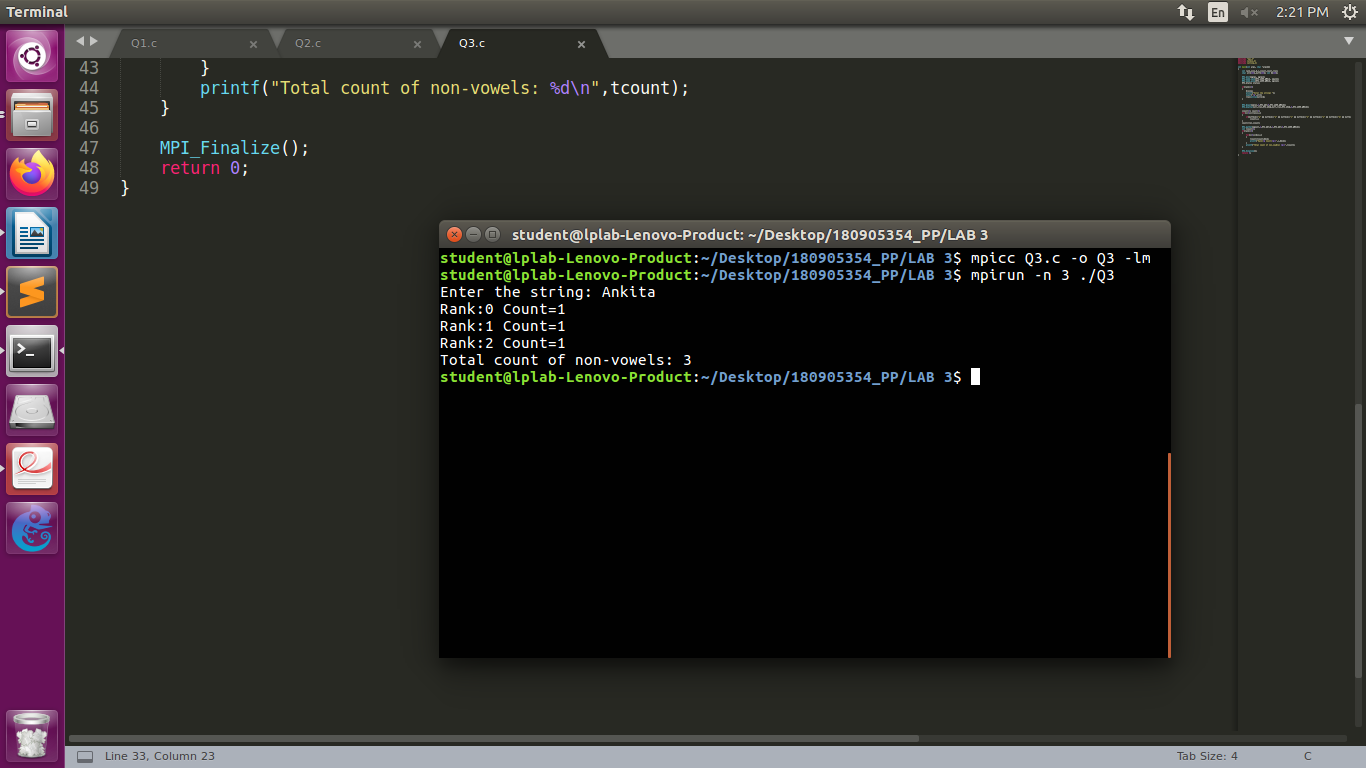
}

MPI\_Finalize();

return 0;

}

Output:



**Q4.**

Program:  
#include "mpi.h"

#include <stdio.h>

#include <string.h>

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

{

int rank,size,N,i,vcount,count,rlen;

char str1[100],str2[100],buff1[100],buff2[100],fbuf[100],strfinal[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)

{

N=size;

printf("Enter first string: ");

scanf("%s",str1);

printf("Enter second string: ");

scanf("%s",str2);

rlen=strlen(str1)/N;

}

MPI\_Bcast(&rlen,1,MPI\_INT,0,MPI\_COMM\_WORLD);

MPI\_Scatter(str1,rlen,MPI\_CHAR,buff1,rlen,MPI\_CHAR,0,MPI\_COMM\_WORLD);

MPI\_Scatter(str2,rlen,MPI\_CHAR,buff2,rlen,MPI\_CHAR,0,MPI\_COMM\_WORLD);

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

{

fbuf[i\*2]=buff1[i];

fbuf[(i\*2)+1]=buff2[i];

}

MPI\_Gather(fbuf,rlen\*2,MPI\_CHAR,strfinal,rlen\*2,MPI\_CHAR,0,MPI\_COMM\_WORLD);

int tcount=0;

if(rank==0)

{

printf("Final string: ");

puts(strfinal);

}

MPI\_Finalize();

return 0;

}

Output:

