Name: Ankita Ghosh Sec: A Rollno: 41

Regno: 180905354 Sem: VI Branch: CSE

PARALLEL PROGRAMMING LAB 1

**Q1**.

Program:

#include <stdio.h>

#include "mpi.h"

#include <stdlib.h>

#include <stdlib.h>

#include <math.h>

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

{

int rank, size, x=5;

MPI\_Init(&argc, &argv);

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

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

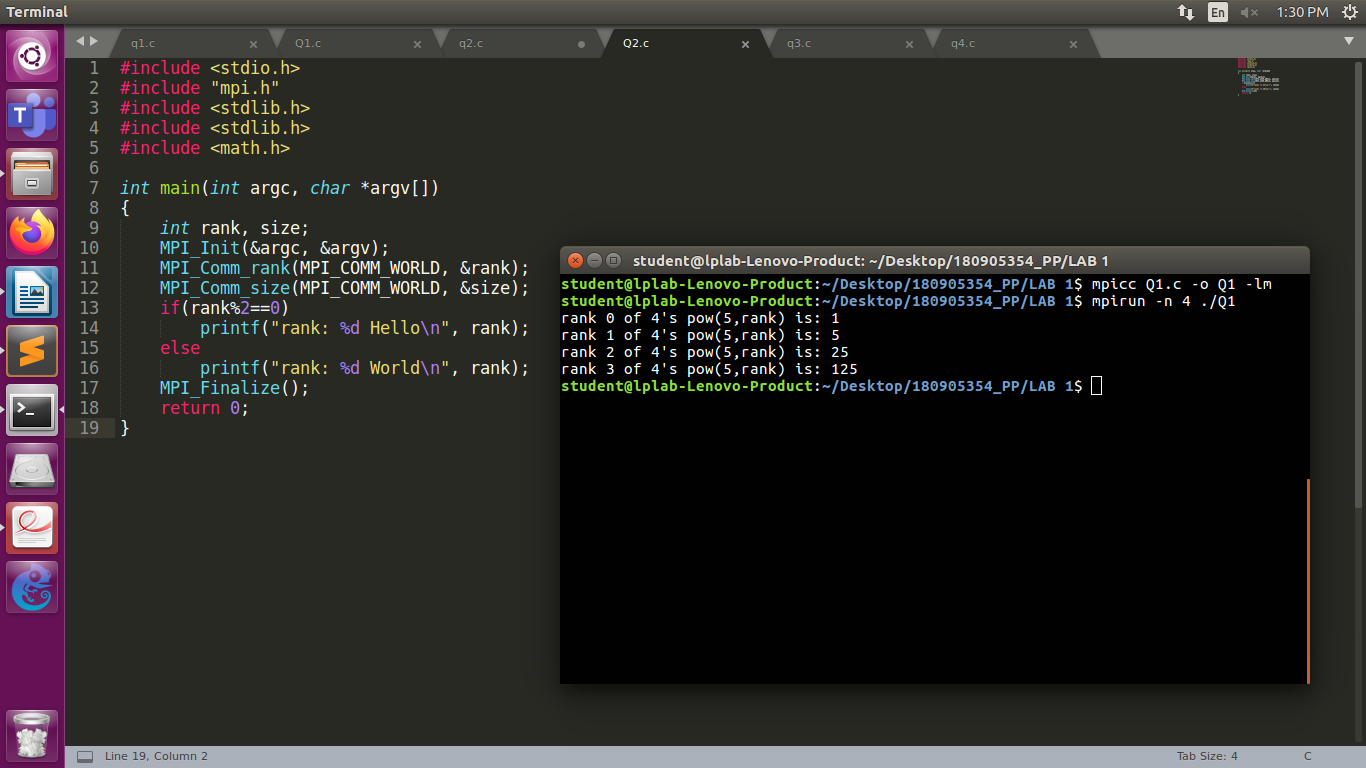
int power = pow(x,rank);

printf("rank %d of %d's pow(%d,rank) is: %d\n", rank, size, x, power);

MPI\_Finalize();

return 0;

}  
  
Output:



**Q2**.

Program:

#include <stdio.h>

#include "mpi.h"

#include <stdlib.h>

#include <stdlib.h>

#include <math.h>

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

{

int rank, size;

MPI\_Init(&argc, &argv);

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

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

if(rank%2==0)

printf("rank: %d Hello\n", rank);

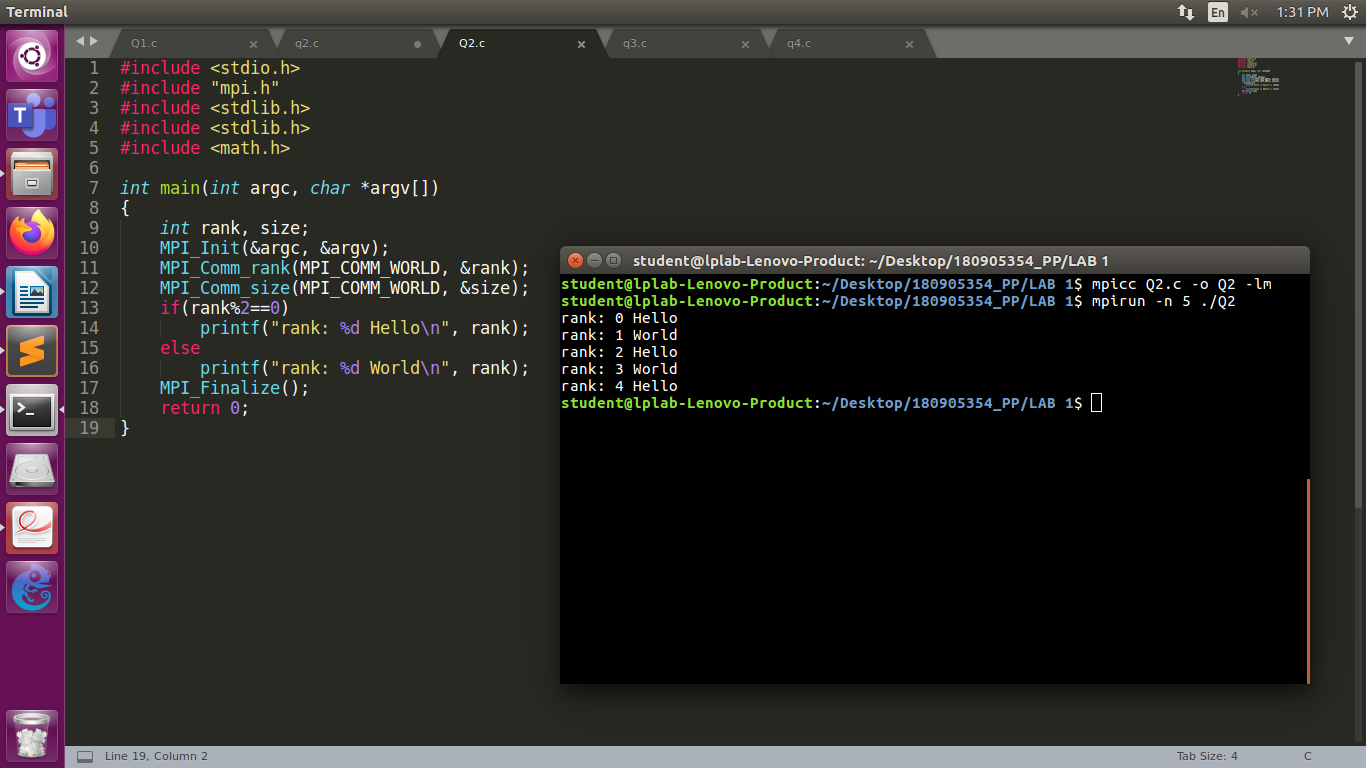
else

printf("rank: %d World\n", rank);

MPI\_Finalize();

return 0;

}  
  
Output:



**Q3**.

Program:

#include <stdio.h>

#include "mpi.h"

#include <stdlib.h>

#include <stdlib.h>

#include <math.h>

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

{

int x=10;

int y=2;

int rank, size;

MPI\_Init(&argc, &argv);

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

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

if(rank==0)

{

printf("rank:%d ADDITION=%d\n", rank, x+y);

}

else if(rank==1)

{

printf("rank:%d SUBTRACTION=%d\n", rank, x-y);

}

else if(rank==2)

{

printf("rank:%d MULTIPLICATION=%d\n", rank, x\*y);

}

else if(rank==3)

{

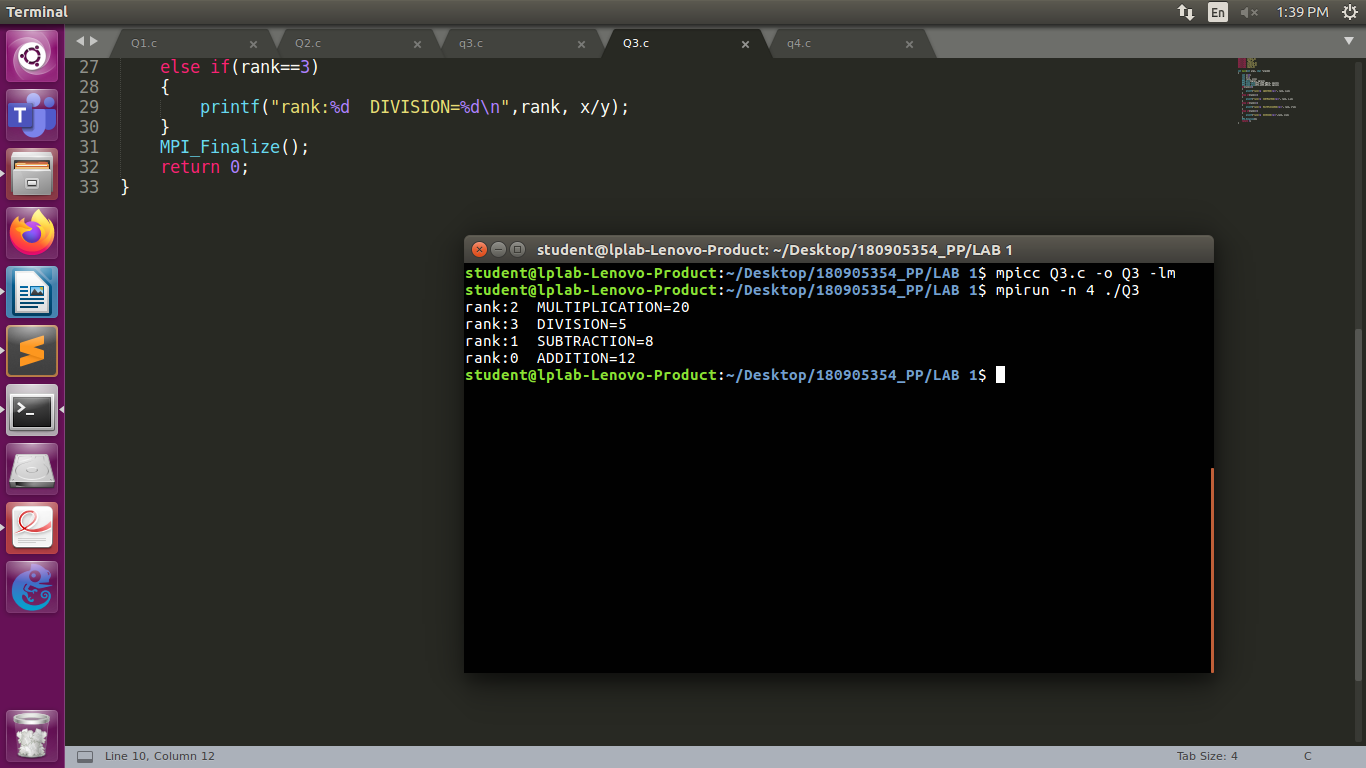
printf("rank:%d DIVISION=%d\n",rank, x/y);

}

MPI\_Finalize();

return 0;

}  
  
Output:



**Q4**.

Program:

#include <stdio.h>

#include "mpi.h"

#include <stdlib.h>

#include <stdlib.h>

#include <math.h>

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

{

int rank, size;

char str[]="Hello";

MPI\_Init(&argc, &argv);

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

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

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

str[rank] = str[rank]+32;

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

str[rank] = str[rank]-32;

printf("Rank: %d String: %s\n", rank, str);

MPI\_Finalize();

return 0;

}  
  
Output:

