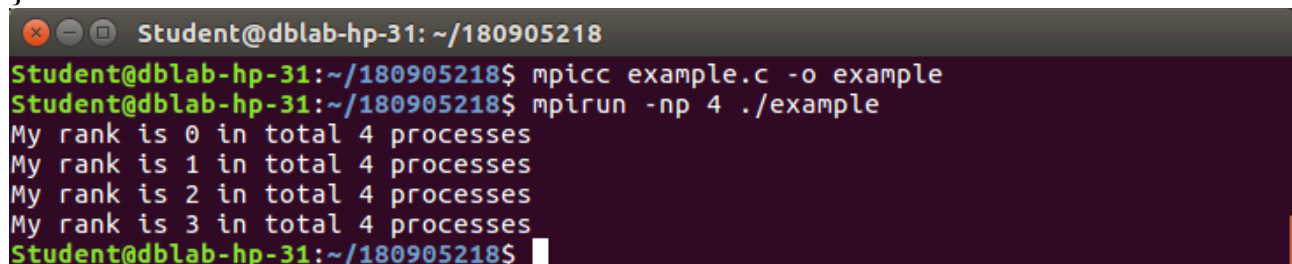


NAME: ARUNIMA SINGH THAKUR
ROLL NO: 31
SECTION: C
LAB: PARALLEL PROGRAMMING
LAB NO: 1
BRANCH: COMPUTER SCIENCE
REGISTRATION NO: 180905218

EXAMPLE:

Write a program in MPI to print total number of process and rank of each process.

```
#include "mpi.h"
#include <stdio.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);
    printf("My rank is %d in total %d processes\n", rank, size);
    MPI_Finalize();
    return 0;
}
```



QUESTION 1:

Write a simple MPI program to find out pow(x,rank) for all the processes where 'x' is the integer constant and 'rank' is the rank of the process.

```
#include "mpi.h"
#include <stdio.h>
#include <math.h>
int main(int argc, char *argv[])
{
    int rank;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
```

```

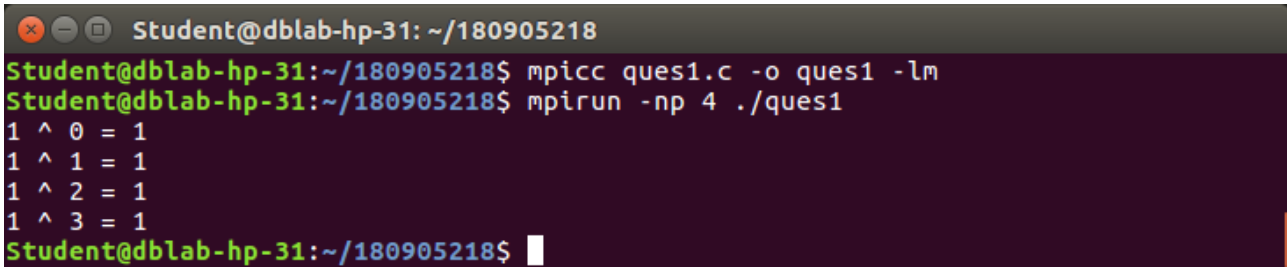
int x=1;

printf("%d ^ %d = %d\n",x,rank,(int)pow(x,rank));

MPI_Finalize();
return 0;

}

```



```

Student@dblab-hp-31: ~/180905218
Student@dblab-hp-31:~/180905218$ mpicc ques1.c -o ques1 -lm
Student@dblab-hp-31:~/180905218$ mpirun -np 4 ./ques1
1 ^ 0 = 1
1 ^ 1 = 1
1 ^ 2 = 1
1 ^ 3 = 1
Student@dblab-hp-31:~/180905218$

```

QUESTION 2:

Write a program in MPI where even ranked process prints “Hello” and odd ranked process prints “World”.

```

#include "mpi.h"
#include <stdio.h>
int main(int argc, char *argv[])
{
    int rank;
    MPI_Init(&argc,&argv);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);

    if(rank%2==0)
    {
        printf("Hello, Rank: %d\n",rank);
    }
    else
    {
        printf("World, Rank: %d\n",rank);
    }

    MPI_Finalize();
    return 0;
}

```

```

Student@dblab-hp-31: ~/180905218
Student@dblab-hp-31:~/180905218$ mpicc ques2.c -o ques2
Student@dblab-hp-31:~/180905218$ mpirun -np 4 ./ques2
World, Rank: 1
Hello, Rank: 2
Hello, Rank: 0
World, Rank: 3
Student@dblab-hp-31:~/180905218$

```

QUESTION 3:

Write a program in MPI to simulate simple calculator. Perform each operation using different process in parallel.

```

#include "mpi.h"
#include <stdio.h>
int main(int argc, char *argv[])
{
    int rank;
    MPI_Init(&argc,&argv);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);

    int x=5;
    int y=6;

    if(rank==0)
        printf("Addition: %d + %d = %d\n",x,y,(x+y));
    else if(rank==1)
        printf("Subtraction: %d - %d = %d\n",x,y,(x-y));
    else if(rank==2)
        printf("Multiplication: %d * %d = %d\n",x,y,(x*y));
    else if(rank==3)
        printf("Division: %d / %d = %d\n",x,y,(x/y));

    MPI_Finalize();
    return 0;
}

```

```

Student@dblab-hp-31: ~/180905218
Student@dblab-hp-31:~/180905218$ mpicc ques3.c -o ques3
Student@dblab-hp-31:~/180905218$ mpirun -np 4 ./ques3
Subtraction: 5 - 6 = -1
Multiplication: 5 * 6 = 30
Division: 5 / 6 = 0
Addition: 5 + 6 = 11
Student@dblab-hp-31:~/180905218$

```

QUESTION 4:

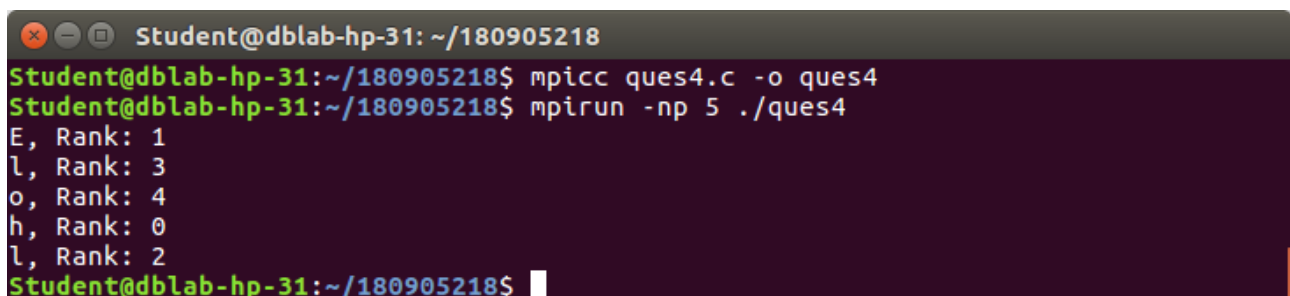
Write a program in MPI to toggle the character of a given string indexed by the rank of the process. HINT: Suppose the string is HeLLO and there are 5 processes, then process 0 toggle 'H' to 'h', process 1 toggle 'e' to 'E' and so on.

```
#include "mpi.h"
#include <stdio.h>
#include <string.h>
int main(int argc, char *argv[])
{
    int rank;
    char str[5]="HeLLO";
    MPI_Init(&argc,&argv);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);

    if(rank< strlen(str))
    {
        if(str[rank]>='A' && str[rank]<='Z')
            str[rank]+=32;
        else if(str[rank]>='a' && str[rank]<='z')
            str[rank]-=32;

        printf("%c, Rank: %d\n",str[rank],rank);
    }

    MPI_Finalize();
    return 0;
}
```



```
Student@dblab-hp-31: ~/180905218
Student@dblab-hp-31:~/180905218$ mpicc ques4.c -o ques4
Student@dblab-hp-31:~/180905218$ mpirun -np 5 ./ques4
E, Rank: 1
l, Rank: 3
o, Rank: 4
h, Rank: 0
l, Rank: 2
Student@dblab-hp-31:~/180905218$
```

ADDITIONAL QUESTION-2

Write a MPI program to find the prime numbers between 1 and 100 using two processes.

```

#include "mpi.h"
#include <stdio.h>
int main(int argc, char *argv[])
{
    int rank;
    int min;
    int max;
    MPI_Init(&argc,&argv);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);

    int is_prime(int num)
    {
        int flag=0;
        for(int i=1;i<=num;i++)
            if(num%i==0)
                flag++;

        if(flag>2)
            return 0;
        else
            return 1;
    }

    switch(rank)
    {
        case 0: min=2;
                max=50;
                break;
        case 1: min=51;
                max=100;
                break;
    }

    for(int i=min;i<=max;i++)
    {
        if(is_prime(i))
            printf("%d\n",i);
    }

    MPI_Finalize();
    return 0;
}

```

```
Student@dblab-hp-31: ~/180905218
Student@dblab-hp-31:~/180905218$ mpicc ques5-add.c -o ques5-add
Student@dblab-hp-31:~/180905218$ mpirun -np 2 ./ques5-add
2
3
5
7
11
13
17
19
23
29
31
37
53
59
61
67
71
73
79
83
89
97
41
43
47
Student@dblab-hp-31:~/180905218$
```