# Distributed Computing (SS ZG526)

## Assignment –1

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

## Submitted by

## Angelin Mary John (2018HT12675)

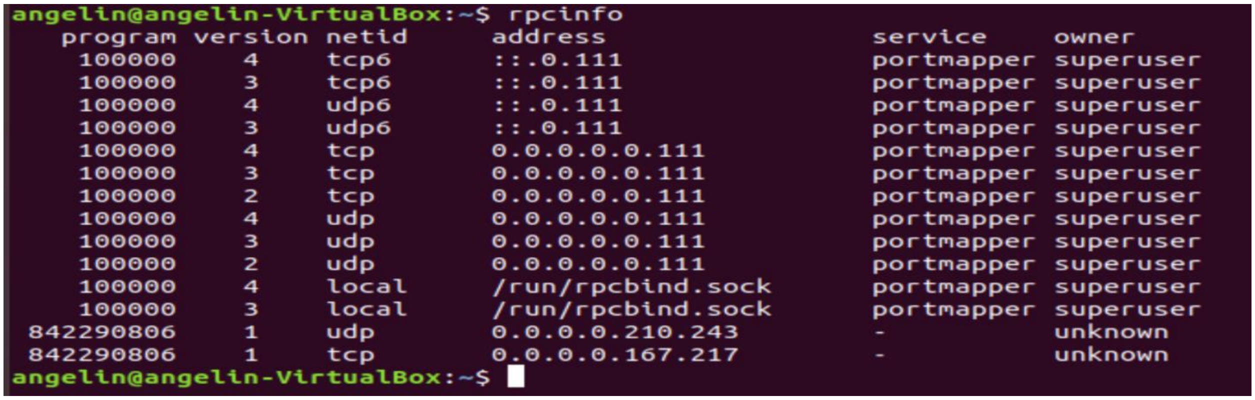
# Prerequisites

The tasks were completed on a linux VM running on a windows OS.

# Task 1- Implementing Calculator

Code is attached in the zip file.

The rpcbind was installed and working ensured using rpcinfo.

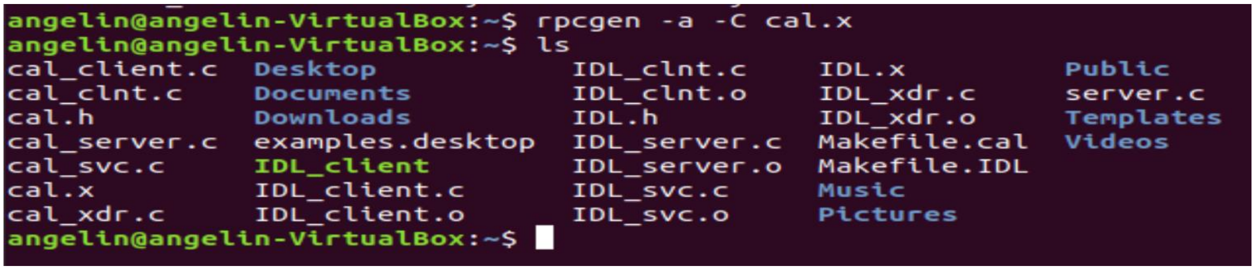


```
angelin@angelin-VirtualBox:~$ rpcinfo
   program version netid     address                service    owner
    100000    4    tcp6      ::.0.111               portmapper superuser
    100000    3    tcp6      ::.0.111               portmapper superuser
    100000    4    udp6      ::.0.111               portmapper superuser
    100000    3    udp6      ::.0.111               portmapper superuser
    100000    4    tcp       0.0.0.0.0.111          portmapper superuser
    100000    3    tcp       0.0.0.0.0.111          portmapper superuser
    100000    2    tcp       0.0.0.0.0.111          portmapper superuser
    100000    4    udp       0.0.0.0.0.111          portmapper superuser
    100000    3    udp       0.0.0.0.0.111          portmapper superuser
    100000    2    udp       0.0.0.0.0.111          portmapper superuser
    100000    4    local     /run/rpcbind.sock      portmapper superuser
    100000    3    local     /run/rpcbind.sock      portmapper superuser
 842290806    1    udp       0.0.0.0.210.243        -          unknown
 842290806    1    tcp       0.0.0.0.167.217        -          unknown
angelin@angelin-VirtualBox:~$ 
```

Using the cal.x IDL file and the rpcgen command, the following files were generated:

Rpcgen command : rpcgen -a -C cal.x

1. Client stub
2. Server skeleton
3. Sample client program
4. Sample server program
5. Header file
6. XDR routines used by both the client and the server.
7. Makefile

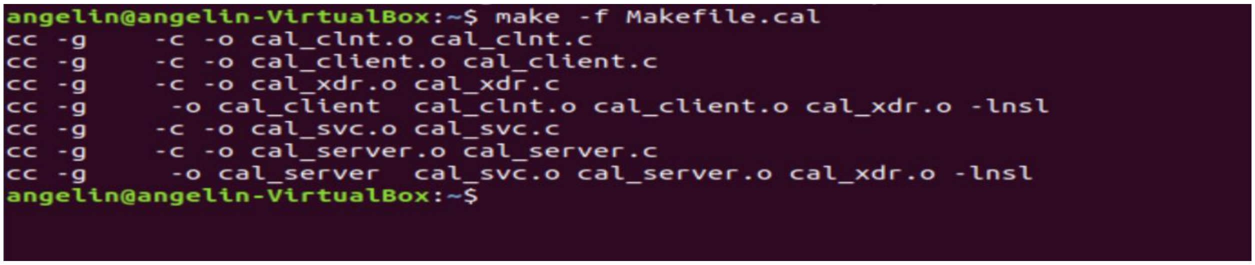


```
angelin@angelin-VirtualBox:~$ rpcgen -a -C cal.x
angelin@angelin-VirtualBox:~$ ls
cal_client.c  Desktop           IDL_clnt.c    IDL.x         Public
cal_clnt.c    Documents         IDL_clnt.o    IDL_xdr.c     server.c
cal.h         Downloads         IDL.h         IDL_xdr.o     Templates
cal_server.c  examples.desktop  IDL_server.c  Makefile.cal  Videos
cal_svc.c     IDL_client        IDL_server.o  Makefile.IDL
cal.x         IDL_client.c      IDL_svc.c     Music
cal_xdr.c     IDL_client.o      IDL_svc.o     Pictures
angelin@angelin-VirtualBox:~$ 
```

The cal_client.c and cal_server.c files were modified as per the requirements.

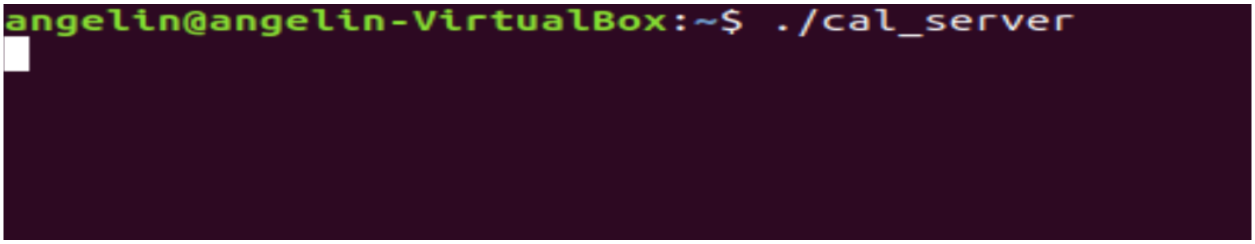
The makefile was recompiled using the command

$ make –f Makefile.cal



```
angelin@angelin-VirtualBox:~$ make -f Makefile.cal
cc -g    -c -o cal_clnt.o cal_clnt.c
cc -g    -c -o cal_client.o cal_client.c
cc -g    -c -o cal_xdr.o cal_xdr.c
cc -g     -o cal_client  cal_clnt.o cal_client.o cal_xdr.o -lnsl
cc -g    -c -o cal_svc.o cal_svc.c
cc -g    -c -o cal_server.o cal_server.c
cc -g     -o cal_server  cal_svc.o cal_server.o cal_xdr.o -lnsl
angelin@angelin-VirtualBox:~$
```

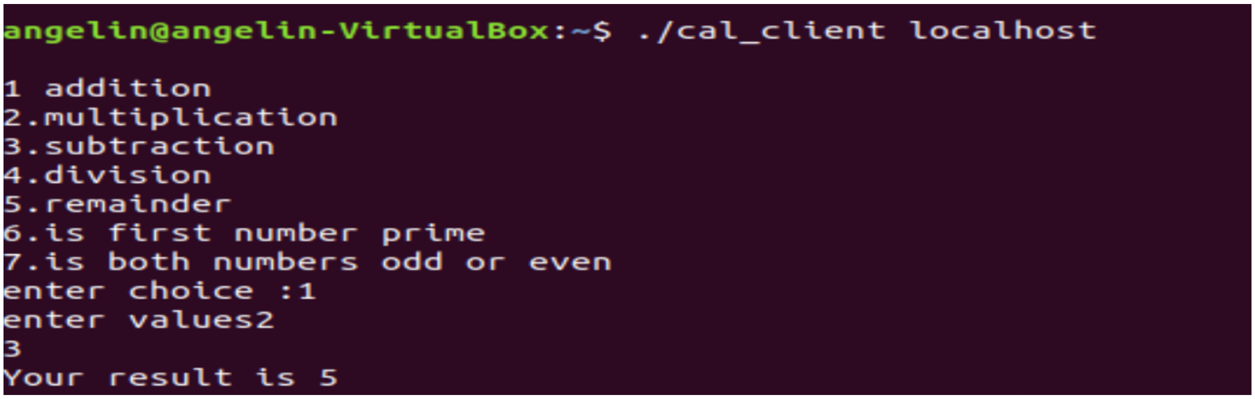
The client and server files were run on two separate terminals and outputs obtained.



```
angelin@angelin-VirtualBox:~$ ./cal_server
```

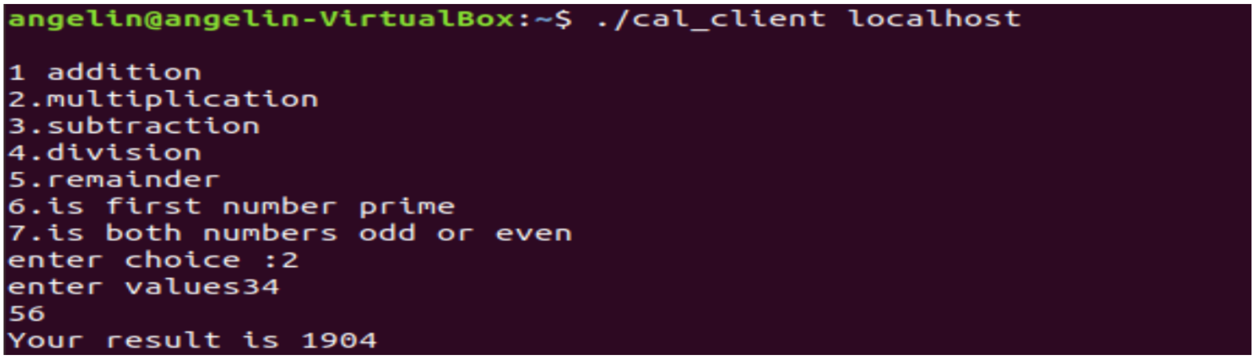
Screen shots of the output are attached below:

1. Addition



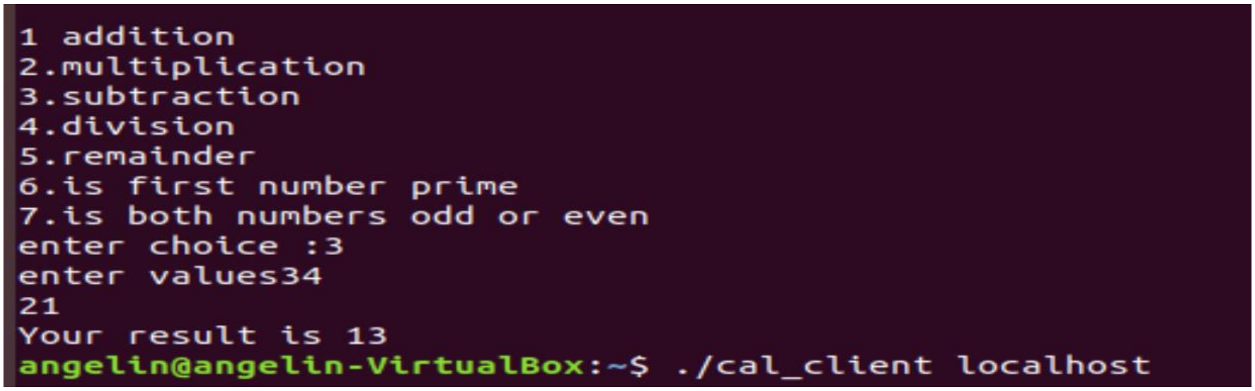
```
angelin@angelin-VirtualBox:~$ ./cal_client localhost

1 addition
2.multiplication
3.subtraction
4.division
5.remainder
6.is first number prime
7.is both numbers odd or even
enter choice :1
enter values2
3
Your result is 5
```

2. Multiplication



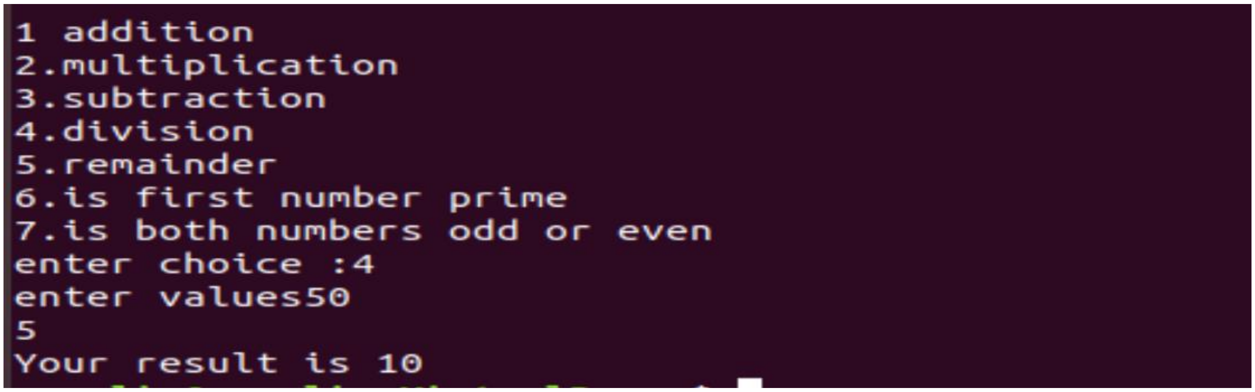
```
angelin@angelin-VirtualBox:~$ ./cal_client localhost

1 addition
2.multiplication
3.subtraction
4.division
5.remainder
6.is first number prime
7.is both numbers odd or even
enter choice :2
enter values34
56
Your result is 1904
```

3. Subtraction



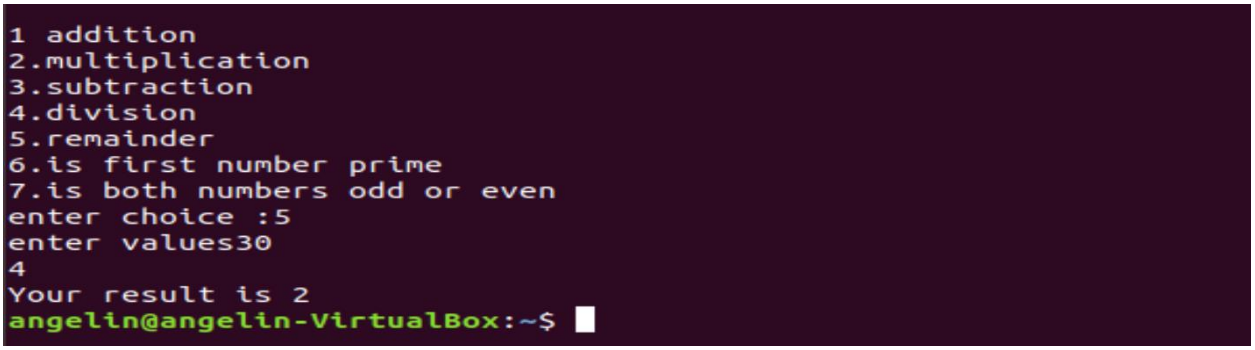
```
1 addition
2.multiplication
3.subtraction
4.division
5.remainder
6.is first number prime
7.is both numbers odd or even
enter choice :3
enter values34
21
Your result is 13
angelin@angelin-VirtualBox:~$ ./cal_client localhost
```

4. Division



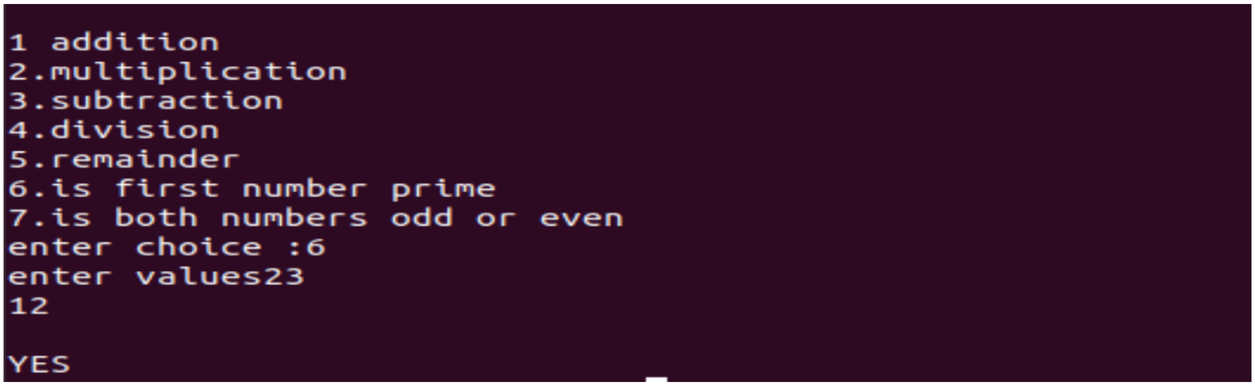
```
1 addition
2.multiplication
3.subtraction
4.division
5.remainder
6.is first number prime
7.is both numbers odd or even
enter choice :4
enter values50
5
Your result is 10
```

5. Remainder



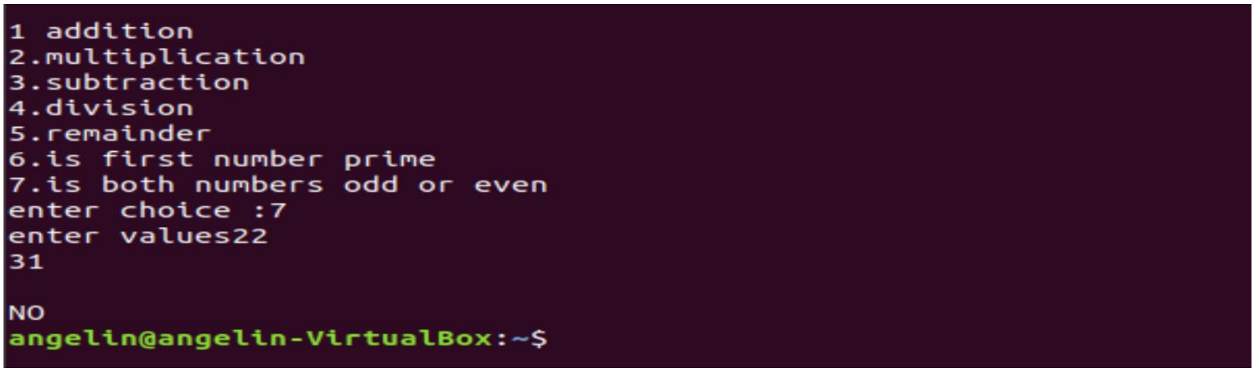
```
1 addition
2.multiplication
3.subtraction
4.division
5.remainder
6.is first number prime
7.is both numbers odd or even
enter choice :5
enter values30
4
Your result is 2
angelin@angelin-VirtualBox:~$
```

6. Is first Number prime



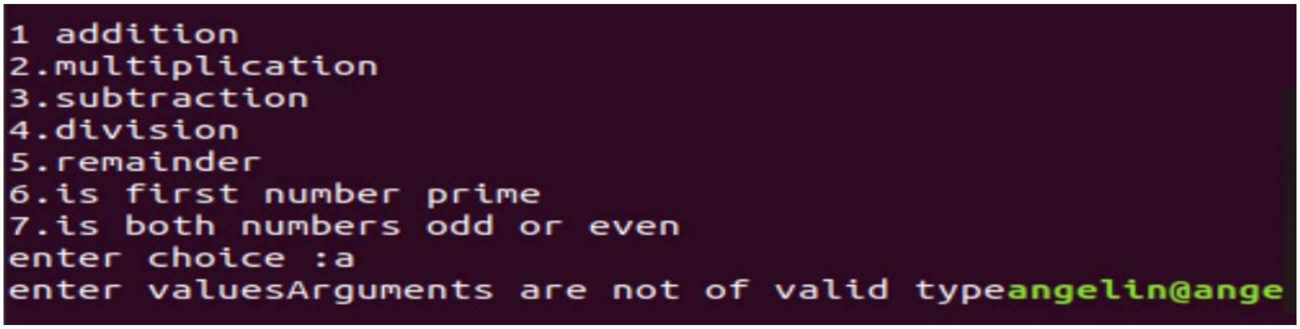
```
1 addition
2.multiplication
3.subtraction
4.division
5.remainder
6.is first number prime
7.is both numbers odd or even
enter choice :6
enter values23
12

YES
```

7. Are both numbers odd or even



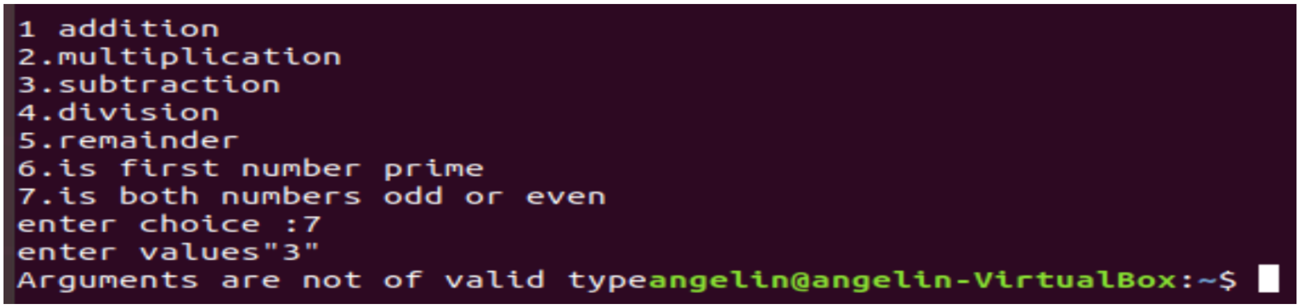
```
1 addition
2.multiplication
3.subtraction
4.division
5.remainder
6.is first number prime
7.is both numbers odd or even
enter choice :7
enter values22
31

NO
angelin@angelin-VirtualBox:~$
```

8. Invalid Choice



```
1 addition
2.multiplication
3.subtraction
4.division
5.remainder
6.is first number prime
7.is both numbers odd or even
enter choice :a
enter valuesArguments are not of valid typeangelin@ange
```

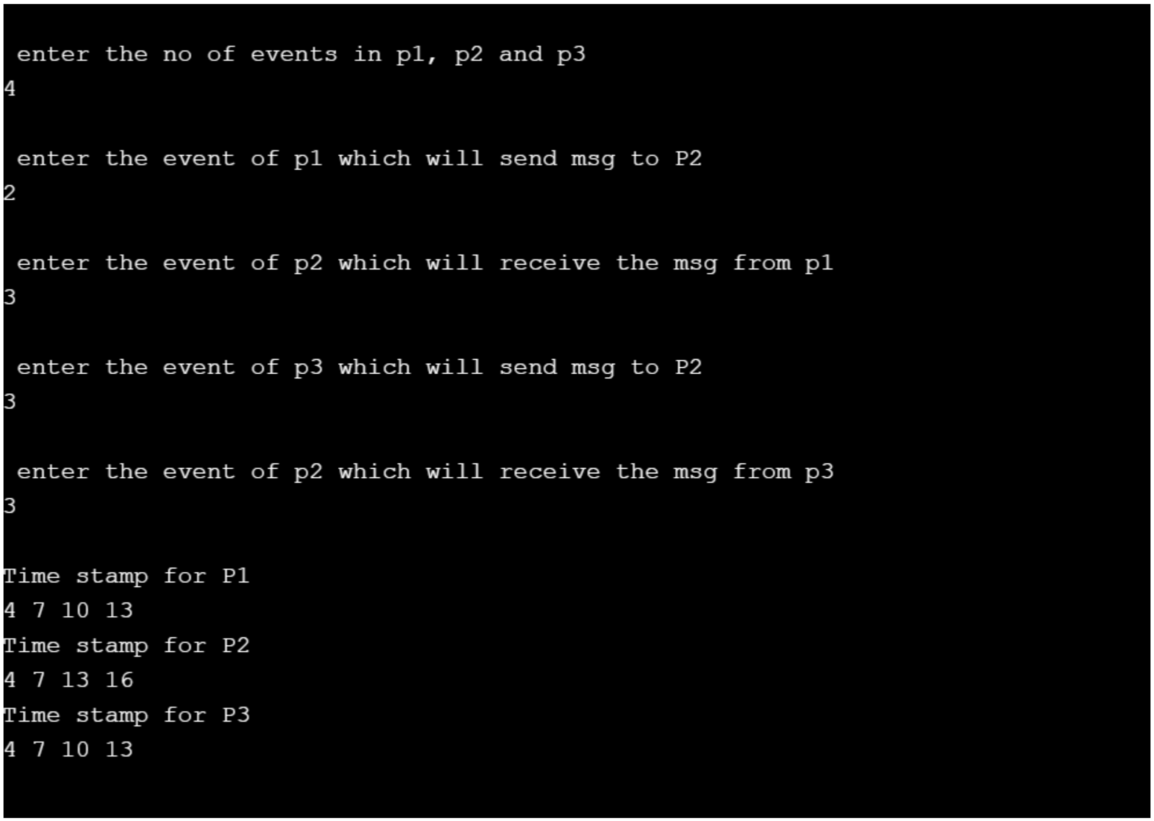
9. Invalid input



```
1 addition
2.multiplication
3.subtraction
4.division
5.remainder
6.is first number prime
7.is both numbers odd or even
enter choice :7
enter values"3"
Arguments are not of valid typeangelin@angelin-VirtualBox:~$ 
```

# Task 2- Implementing Lamports’ Logical Clock

The three processes with initial values 4 and value of d=3 was implemented using the code attached and screenshot is shown below.



```
 enter the no of events in p1, p2 and p3
4

 enter the event of p1 which will send msg to P2
2

 enter the event of p2 which will receive the msg from p1
3

 enter the event of p3 which will send msg to P2
3

 enter the event of p2 which will receive the msg from p3
3

Time stamp for P1
4 7 10 13
Time stamp for P2
4 7 13 16
Time stamp for P3
4 7 10 13
```