DISTRIBUTED SYSTEMS LAB FILE

BCE – C661

SUBMITTED BY:	SUBMITTED TO:
ANKIT BANSAL	Mr. Sumit Bansal
Group-A	Assistant Professor
B.TECH, CSE, VI SEM	CSE Department, FET, GKV

DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING FACULTY OF ENGINEERING AND
TECHNOLOGY
GURUKUL KANGRI UNIVERSITY 2022-2023

INDEX

S.NO.	NAME OF PRACTICAL	SIGNATURE
1.	Write a program in c++ for the implementation of non-token-based algorithms for distributed mutual exclusion.	
2.	Write a program in c++ to implement Lamport's logical clock.	
3.	Write a program in c++ to implement edge chasing distributed deadlock detection algorithms.	
4.	Write a program in c++ to implement a locking algorithm.	
5.	Write a program to implement Remote Method Invocation	
6.	Write a program to implement Remote Procedure Call	
7.	Create a LAN network and share the resources.	
8.	Remotely Access any computer of LAN Network without using any software.	

Practical: 1

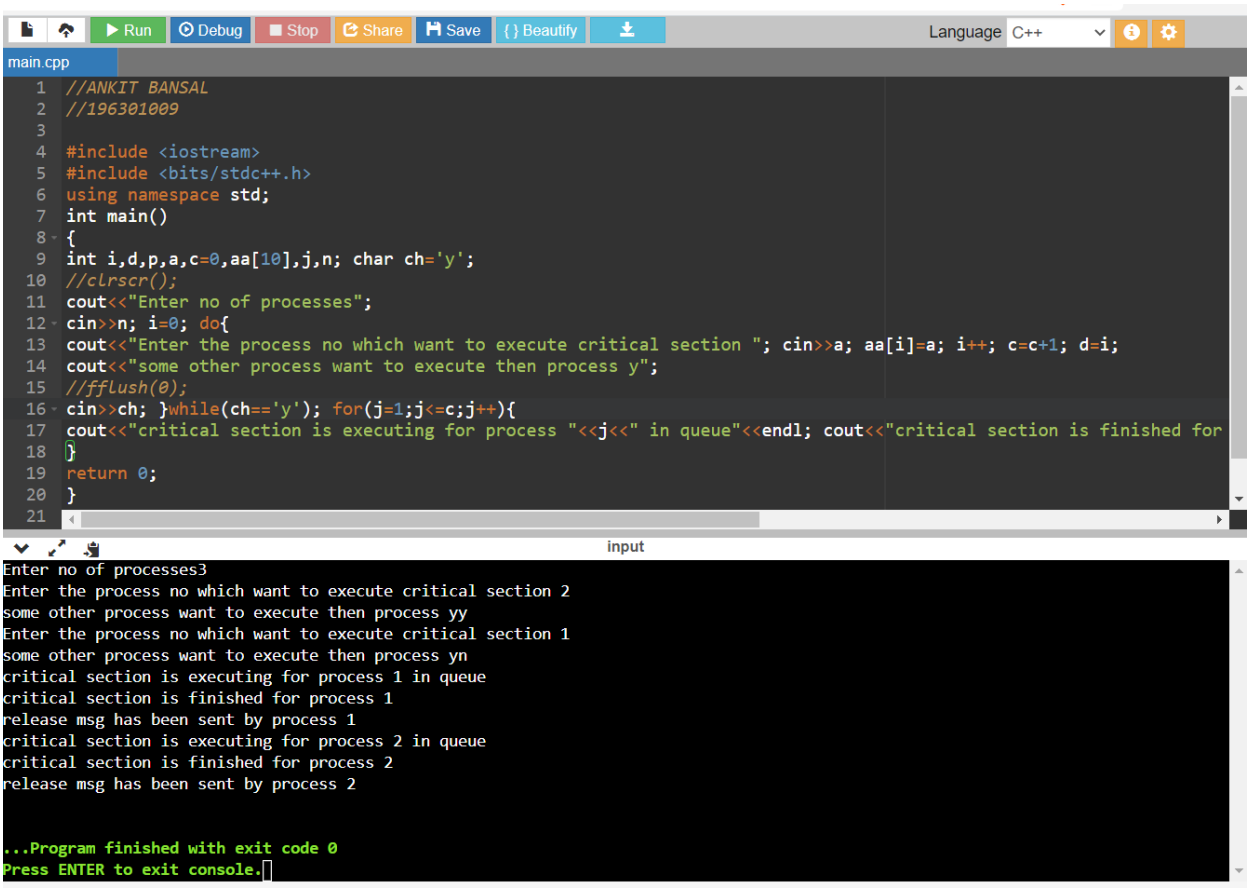
Write a program in c++ for the implementation of non-token-based algorithms for distributed mutual exclusion.

CODE:

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;


int main()
{
int i,d,p,a,c=0,aa[10],j,n;
char ch='y';
//clrscr();
cout<<"Enter no of processes";
cin>>n;
i=0;
do{
cout<<"Enter the process no which want to execute critical section ";
cin>>a; aa[i]=a; i++; c=c+1; d=i;
cout<<"some other process want to execute then process y";
//fflush(0);
cin>>ch;
}while(ch=='y');
for(j=1;j<=c;j++){
cout<<"critical section is executing for process "<<j<<" in queue"<<endl;
cout<<"critical section is finished for process "<<j<<endl;
cout<<"release msg has been sent by process "<<j<<endl;
}
return 0;
}
```

ANKIT BANSAL
196301009

OUTPUT:



Practical: 2

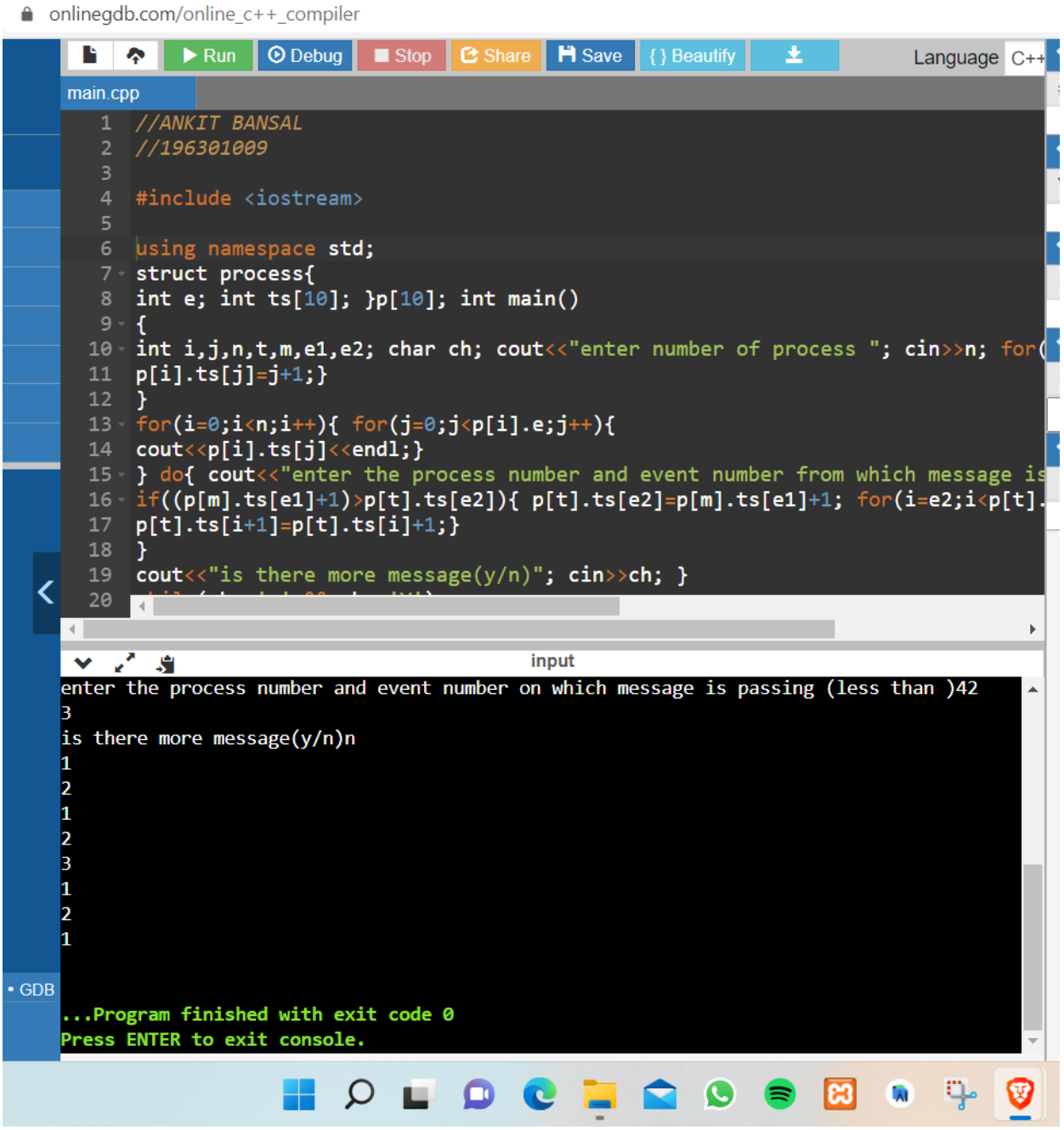
Write a program in c++ to implement Lamport's logical clock.

CODE:

```
#include <iostream>
using namespace std;
struct process{
int e; int
ts[10];
}p[10];
int
main()
{
int i,j,n,t,m,e1,e2; char ch; cout<<"enter number
of process "; cin>>n; for(i=0;i<n;i++){
cout<<"enter number of events in process
"<<i+1; cin>>p[i].e; for(j=0;j<p[i].e;j++){
p[i].ts[j]=j+1;}
}
```

```
for(i=0;i<n;i++){
for(j=0;j<p[i].e;j++){
cout<<p[i].ts[j]<<endl;}
} do{ cout<<"enter the process number and event number from which message is passing(less than )"<<n; cin>>m>>e1; cout<<"enter the process number and event number on which message is passing (less than )"<<n; cin>>t>>e2;
if((p[m].ts[e1]+1)>p[t].ts[e2]){
p[t].ts[e2]=p[m].ts[e1]+1;
for(i=e2;i<p[t].e;i++){
p[t].ts[i+1]=p[t].ts[i]+1;}
}
cout<<"is there more message(y/n)";
cin>>ch; }
while(ch=='y' && ch=='Y');
for(i=0;i<n;i++){
for(j=0;j<p[i].e;j++){
cout<<p[i].ts[j]<<endl;}
}
return 0;}
```

OUTPUT:



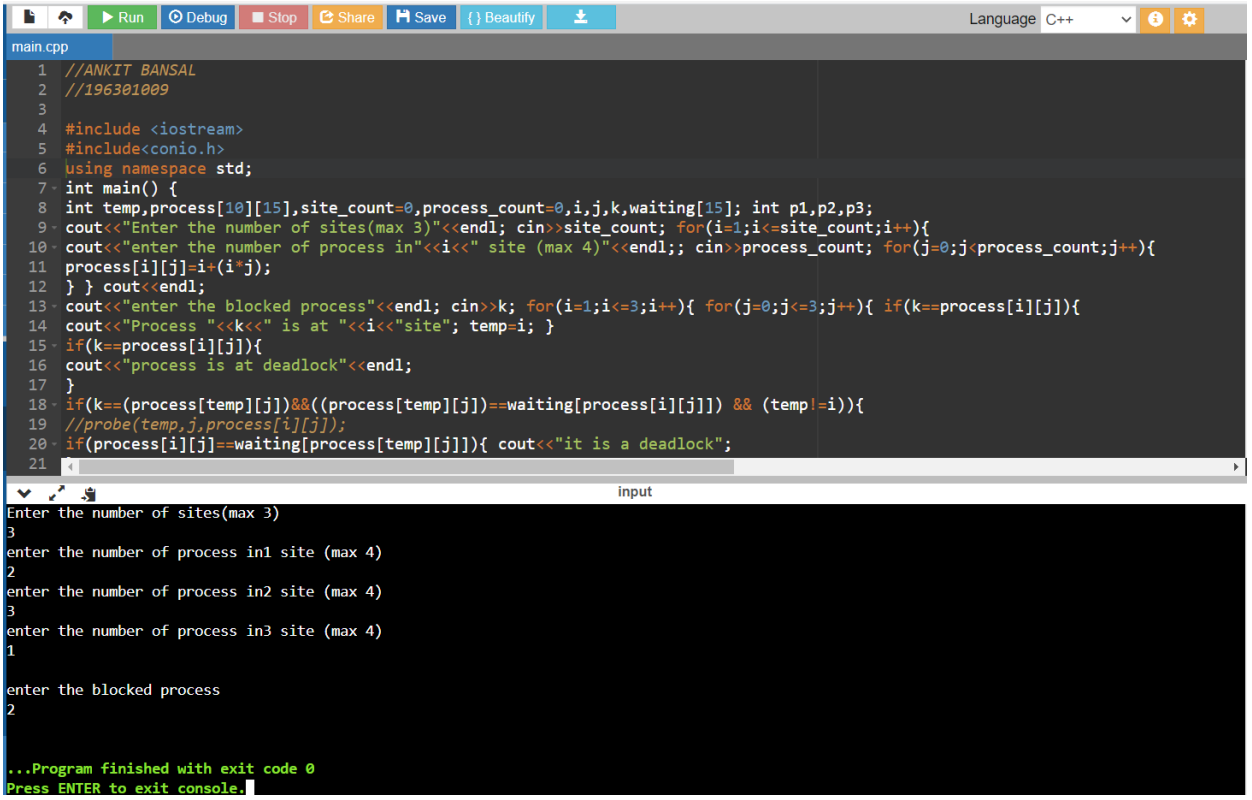
Practical: 3

Write a program in c++ to implement edge chasing distributed deadlock detection algorithms.

CODE:

```
#include <iostream>
#include<conio.h>
using namespace std;


int main()
{
int temp,process[10][15],site_count=0,process_count=0,i,j,k,waiting[15];
int p1,p2,p3;
cout<<"Enter the number of sites(max
3)"<<endl; cin>>site_count;
for(i=1;i<=site_count;i++){
cout<<"enter the number of process in"<<i<<" site (max 4)"<<endl;;
cin>>process_count; for(j=0;j<process_count;j++){
process[i][j]=i+(i*j);
} }
cout<<endl
;
cout<<"enter the blocked process"<<endl;
cin>>k; for(i=1;i<=3;i++){
for(j=0;j<=3;j++){ if(k==process[i][j]){
cout<<"Process "<<k<<" is at "<<i<<"site";
temp=i; }
if(k==process[i][j]){
cout<<"process is at deadlock"<<endl;
}
if(k==(process[temp][j])&&((process[temp][j])==waiting[process[i][j]]) && (temp!=i)){
//probe(temp,j,process[i][j]);
if(process[i][j]==waiting[process[temp][j]]){
cout<<"it is a deadlock";
}
}
}
}
return 0;
}
```

OUTPUT:



Practical: 4

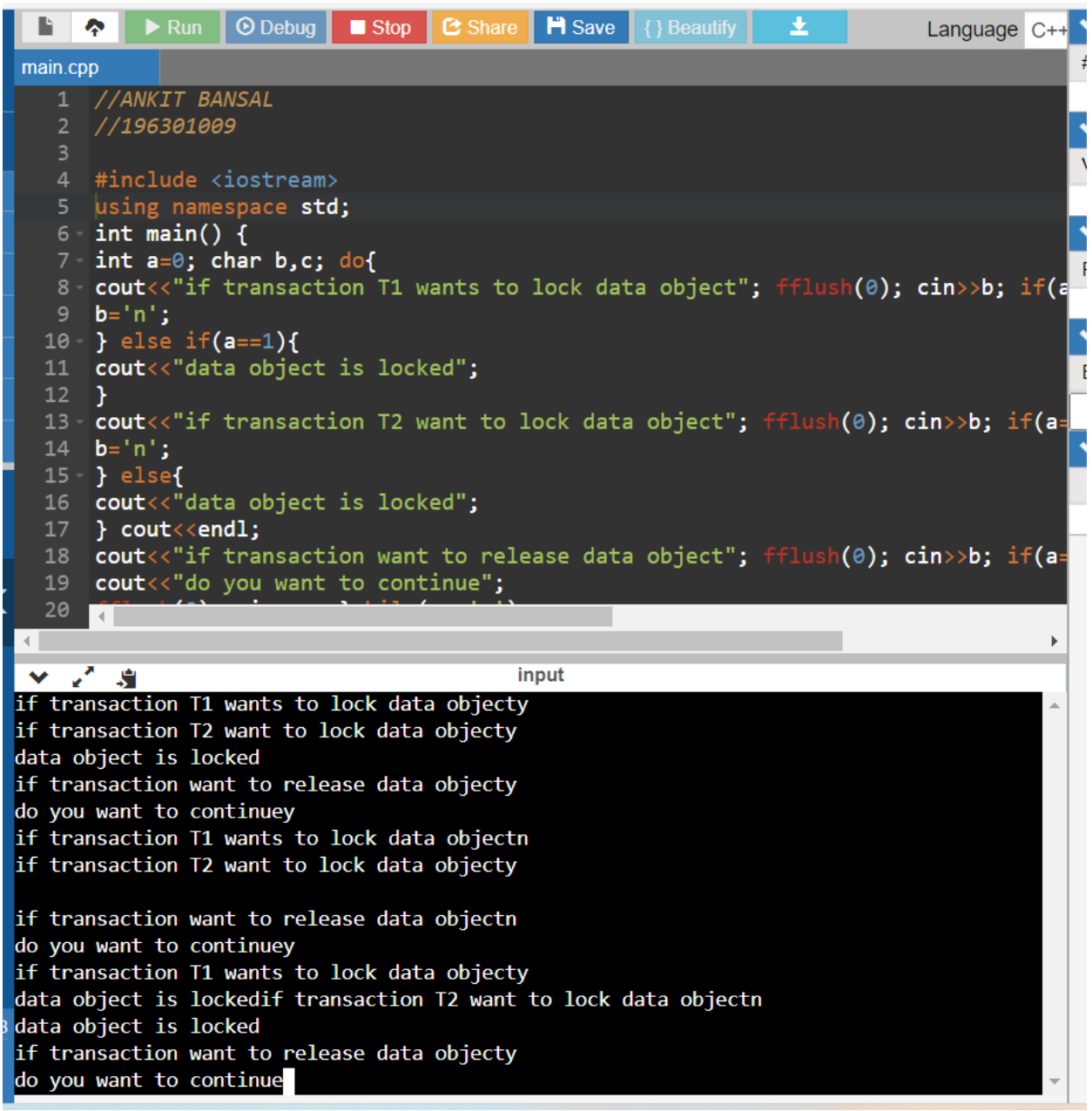
Write a program in c++ to implement a locking algorithm.

CODE:

```
#include <iostream>
using namespace std;


int main()
{
int a=0;
char b,c;
do{
cout<<"if transaction T1 wants to lock data
object"; fflush(0); cin>>b; if(a==0 && b=='y'){
a=1;
b='n';
} else
if(a==1){
cout<<"data object is locked";
}
```

```
cout<<"if transaction T2 want to lock data
object"; fflush(0); cin>>b; if(a==0 && b=='y'){
a=1;
b='n';
}
else{
cout<<"data object is locked";
}
cout<<endl;
cout<<"if transaction want to release data
object"; fflush(0); cin>>b; if(a==1 && b=='y'){
a=0; }
cout<<"do you want to continue";
fflush(0);
cin>>c;
}while(c=='y');
return 0;
}
```

OUTPUT:



Practical: 5

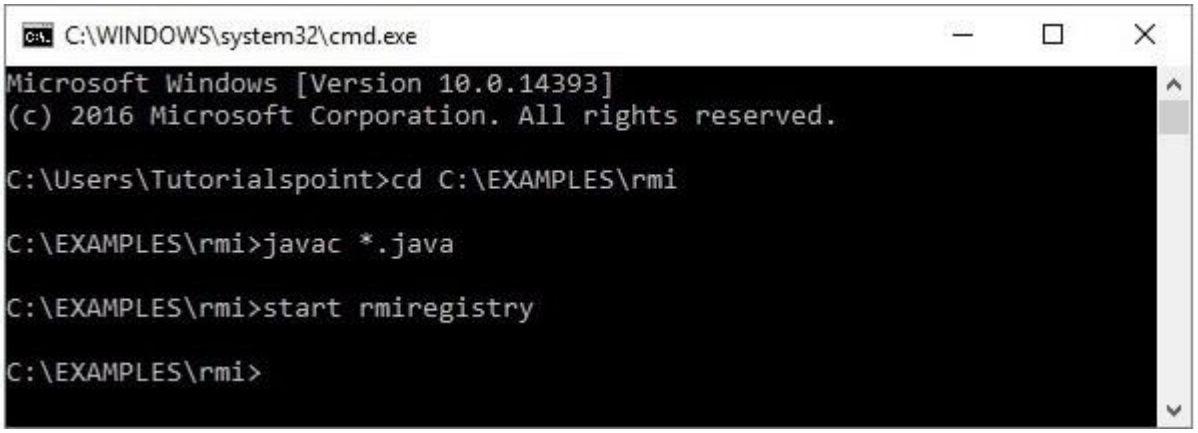
Write a program to implement Remote Method Invocation

```
// Implementing the remote interface public
class ImplExample implements Hello {

  // Implementing the interface method
  public void printMsg() {
    System.out.println("This is an example RMI program");
  }
```

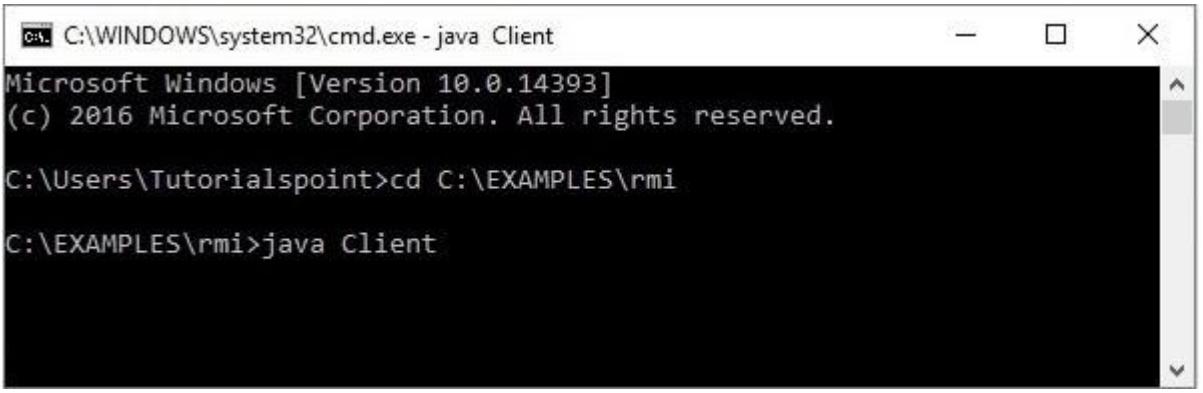
```
}

//Server import

java.rmi.registry.Registry; import

java.rmi.registry.LocateRegistry;

import java.rmi.RemoteException; import
java.rmi.server.UnicastRemoteObject;

public class Server extends ImplExample {
   public Server() {}
   public static void main(String args[]) {
      try {
         // Instantiating the implementation class
         ImplExample obj = new ImplExample();

         // Exporting the object of implementation class
         // (here we are exporting the remote object to the stub)
         Hello stub = (Hello) UnicastRemoteObject.exportObject(obj, 0);

         // Binding the remote object (stub) in the registry
         Registry registry = LocateRegistry.getRegistry();

         registry.bind("Hello", stub);
         System.err.println("Server ready");
      } catch (Exception e) {
         System.err.println("Server exception: " + e.toString());
         e.printStackTrace();
      }
   }
}
//Client import
java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class Client {
   private Client() {} public
   static void main(String[]
   args) {
      try {
         // Getting the registry
         Registry registry = LocateRegistry.getRegistry(null);

         // Looking up the registry for the remote object
         Hello stub = (Hello) registry.lookup("Hello");
```

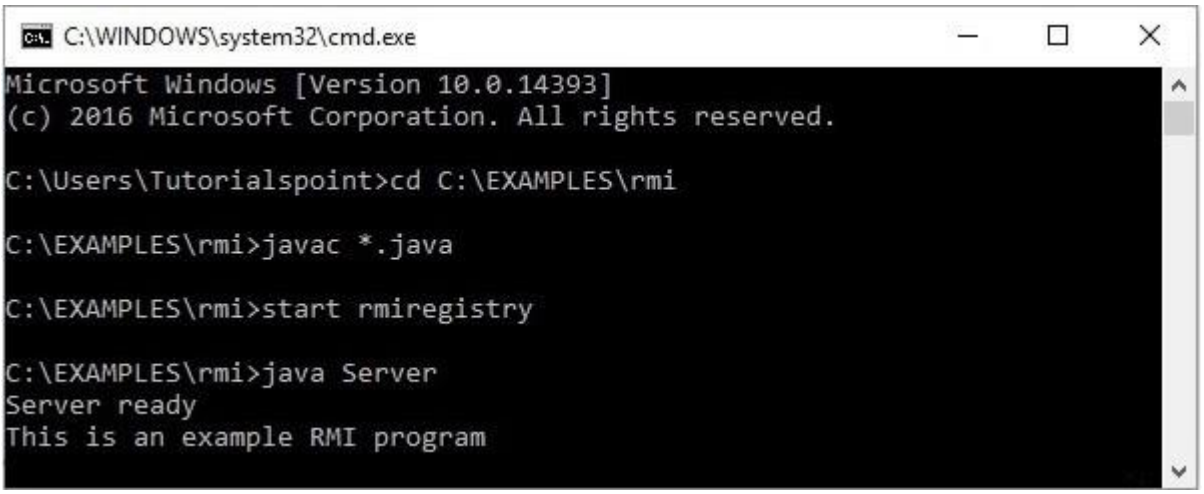
```
        // Calling the remote method using the obtained object
        stub.printMsg();

        // System.out.println("Remote method invoked");
      } catch (Exception e) {
        System.err.println("Client exception: " + e.toString());
        e.printStackTrace();
      }
    }
}
```

OUTPUT:


C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\Users\Tutorialspoint>cd C:\EXAMPLES\rmi
C:\EXAMPLES\rmi>javac *.java
C:\EXAMPLES\rmi>start rmiregistry
C:\EXAMPLES\rmi>


C:\WINDOWS\system32\cmd.exe - java Server
C:\EXAMPLES\rmi>java Server
Server ready


C:\WINDOWS\system32\cmd.exe - java Client
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\Users\Tutorialspoint>cd C:\EXAMPLES\rmi
C:\EXAMPLES\rmi>java Client


C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\Users\Tutorialspoint>cd C:\EXAMPLES\rmi
C:\EXAMPLES\rmi>javac *.java
C:\EXAMPLES\rmi>start rmiregistry
C:\EXAMPLES\rmi>java Server
Server ready
This is an example RMI program

Practical: 6

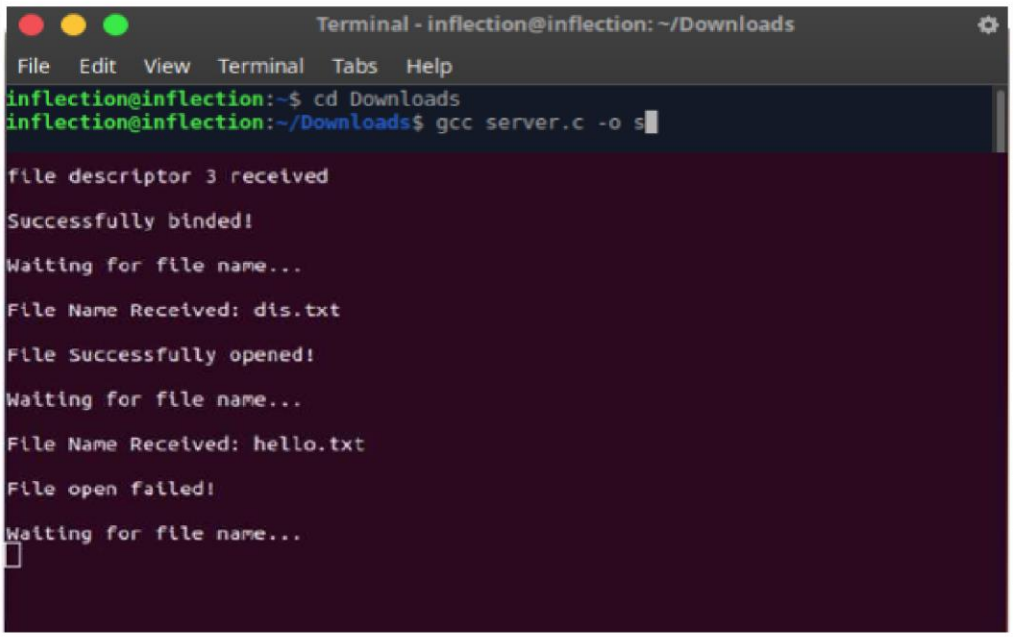
Write a Program to implement Remote Procedure Call.

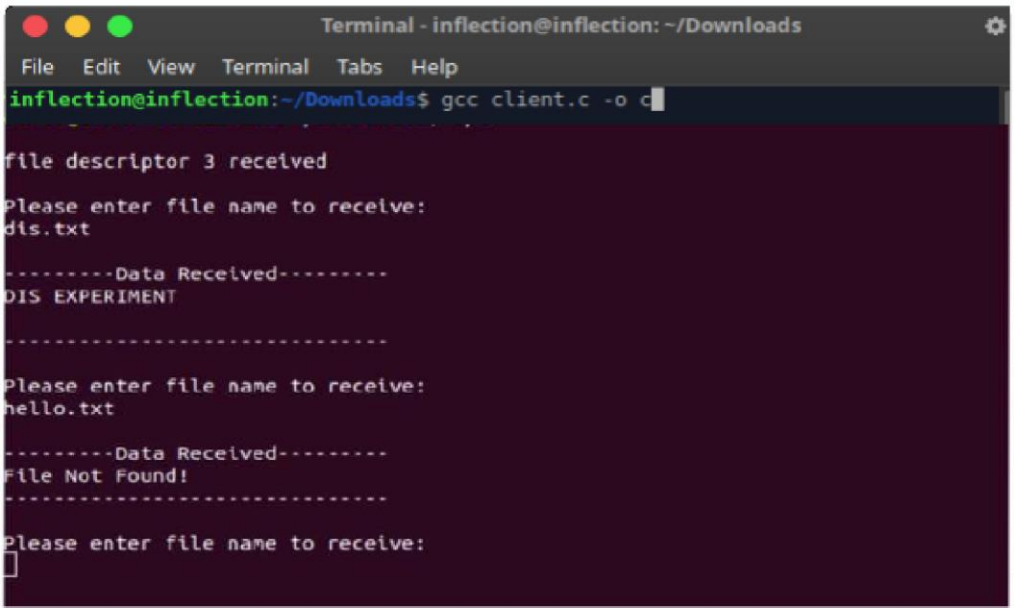
```
include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#define NET_BUF_SIZE 32
#define cipherKey 'S'
#define sendrecvflag 0
#define nofile "File Not Found!"
// function to clear buffer void
clearBuf(char* b)
{
int i;
for (i = 0; i < NET_BUF_SIZE; i++)
b[i] = '\0';
}
// function to encrypt
char Cipher(char ch)
{
return ch ^ cipherKey;
}
// function sending file
int sendFile(FILE* fp, char* buf, int s)
{
int i, len;
if (fp == NULL) {
strcpy(buf, nofile); len
= strlen(nofile);
buf[len] = EOF; for (i
= 0; i <= len; i++)
buf[i] = Cipher(buf[i]);
return 1;
}
char ch, ch2; for (i =
0; i < s; i++) { ch =
fgetc(fp); ch2 =
Cipher(ch); buf[i] =
ch2; if (ch == EOF)
return 1;
}
return 0;
}
// driver code
```

```
int main()
{
int sockfd, nBytes; struct
sockaddr_in addr_con; int
addrlen = sizeof(addr_con);
addr_con.sin_family =
AF_INET; addr_con.sin_port =
htons(PORT_NO);
addr_con.sin_addr.s_addr =
INADDR_ANY; char
net_buf[NET_BUF_SIZE];
FILE* fp; // send
sendto(sockfd, net_buf, NET_BUF_SIZE,
sendrecvflag,
(struct sockaddr*)&addr_con, addrlen);
clearBuf(net_buf); } if (fp != NULL)
fclose(fp);
}
return 0;
}
// client code for UDP socket programming
#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#define IP_PROTOCOL 0
#define IP_ADDRESS "127.0.0.1" // localhost
#define PORT_NO 15050
#define NET_BUF_SIZE 32
#define cipherKey 'S'
#define sendrecvflag 0 //
function to clear buffer
void clearBuf(char* b)
{
int i;
for (i = 0; i < NET_BUF_SIZE; i++)
b[i] = '\0';
}
// function for decryption
char Cipher(char ch) {
return ch ^ cipherKey;
}
// function to receive file
int recvFile(char* buf, int s)
{
int i;
```

```
char ch; for (i = 0; i <
s; i++) { ch = buf[i];
ch = Cipher(ch); if
(ch == EOF) return 1;
else printf("%c", ch);
}
return 0;
}
```

OUTPUT:





Practical: 7

Create a LAN network and share the resources.

1. Gather your equipment

To set up a LAN, you will need:

- A network switch - or a router
- An ethernet cable, plus extra ones for every device you want to connect via cable
- A computer
- All the rest of your devices

If you want your LAN to connect to the internet, you'll also need:

- A broadband connection
- A router
- A modem (if there isn't one built into your router)

Start by plugging in your network switch or router and switching it on.

2. Connect the first computer

Brand new network switch or router? The first thing you need to do is set it up. Do this by connecting it to a computer via an ethernet cable.

On a Windows PC: Using a network switch or router for the first time should bring up the 'Set up a network' wizard - an easy and simple way of getting things set up automatically. If it doesn't appear, or if you've already used this router, go to the Network and Sharing Centre (in the control panel or under the settings) and select 'Set up a new connection or network'. You'll then be taken through the steps.

On a Mac: Go to System Preferences, then Network, Built-In Ethernet, Advanced. This is where you'll find all the settings you need to set up a new network.

3. Set up your Wi-Fi

If you want devices connected to your network wirelessly - the best choice for smartphones, tablets, streaming sticks, and so on - you'll need to set up Wi-Fi (of course, if you only want computers to connect to the LAN via ethernet cable, go ahead and skip this step).

Change the router login password to something unique too - different from the Wi-Fi password, of course,

- Pick the most advanced security tech - currently WPA2.

You may need to restart your computer to see the changes take effect.

4. Connect to the internet

The idea of a LAN is to connect different devices to each other - but if you want it to have internet access too, now's when you need to set it up.
If you already have a working router and broadband connection, you should just be able to plug in and go.
If you're setting up a new router and/or internet connection, on the other hand, you'll need to follow the instructions given to you by your broadband provider or router manufacturer.
Either way, you'll need to plug your router and modem into your home's main phone line, using the router's WAN port.

---

5. Connect the rest of your devices

Whether you're connecting your gadgets to the LAN via Wi-Fi or ethernet cable, the time has come to get it all hooked up. This includes other computers, laptops, smartphones, tablets, TV set top boxes, games consoles, streaming sticks anything that might need to get online.

To connect via Wi-Fi, turn on Wi-Fi on your device, and select your home network from the list. It should be pretty recognizable because you changed the SSID to a custom name, right? You'll then be prompted to enter your new secure password.

6. Get sharing

One of the beauties of a LAN is that you can share resources across it, such as devices, files, and media. With Windows PCs, this is super easy to set up by creating a 'Homegroup'.

Part of setting up your home group involves picking what kinds of files you want to be able to share, but there are plenty of ways to share things on a LAN:

- Right-click on a file or folder, select 'Share with…', and choose who to share it with.
- Or, select it in File Explorer, and go to the 'Share' tab.
- Move files into Public folders - like Public Music, or Public Pictures - and turn on Public Folder Sharing (under the advanced sharing settings in the Network and Sharing Center).
- On a Mac, go to System Preferences, then Sharing preferences, and tick the 'File Sharing checkbox. Click 'Add' to put files in your Public folder for sharing.

Practical: 8

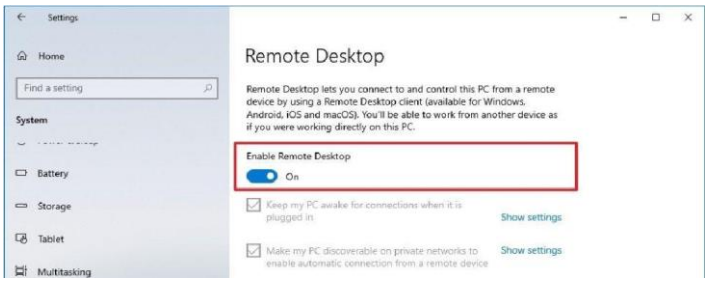
Remotely Access any computer of LAN Network without using any software.

ANKIT BANSAL
196301009

Settings app

To enable the Remote Desktop using the Settings app, use these steps:

1. Open Settings.
2. Click on System.
3. Click on Remote Desktop.
4. Turn on the Enable Remote Desktop toggle switch.
5. Click the Confirm button.

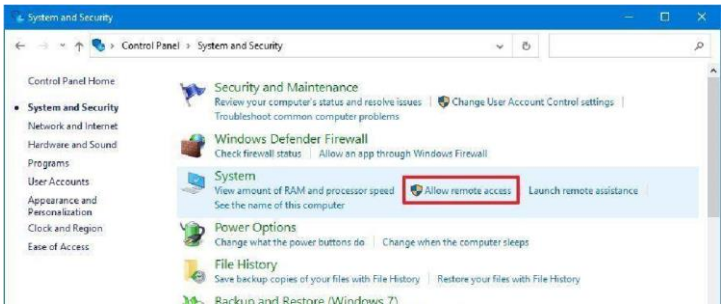


However, you may need to configure the rules manually if you have a third-party firewall.

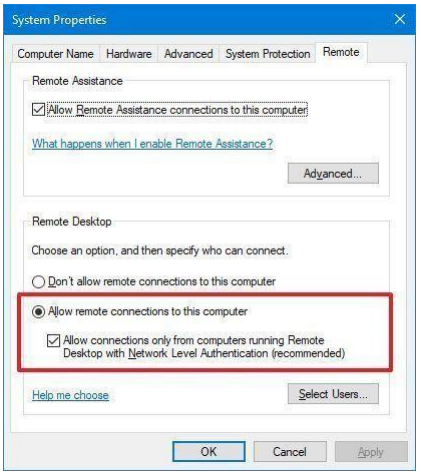
Control Panel

To enable remote connections on Windows 10, use these steps:

1. Open Control Panel.
2. Click on System and Security.
3. Under the "System" section, click the Allow remote access option.



4. Click the Remote tab.
5. Under the "Remote Desktop" section, check the Allow remote connections to this computer option.
6. Check the Allow connections only from computers running Remote Desktop with Network Level Authentication option.
6. Click the OK button.



Enable remote connections on the router

If the remote connection has to happen over an internet connection, you will also have to configure the router to allow the remote connection. In addition, you will need to know the public address to contact the remote computer.

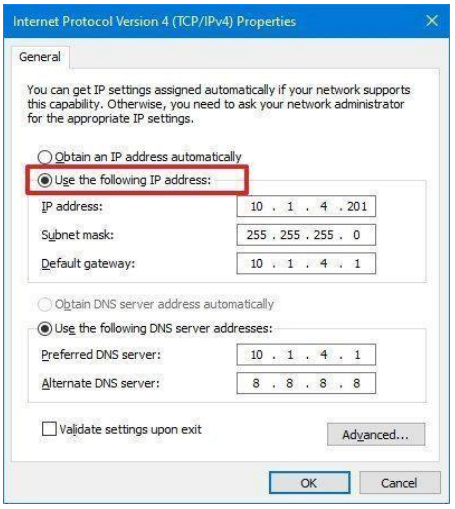
Configure a static IP address on Windows 10

A computer usually receives a dynamic IP address from the DHCP server (router), which means it can change. If you plan to use a remote desktop for a long time, you may want to configure a static IP address to avoid reconfiguring port forwarding on the router when the device changes the network configuration.

Control Panel

To configure a permanent network configuration, use these steps:

1. Open Control Panel.
2. Click on Network and Internet.
3. Click on Network and Sharing Center.
4. Click the Change adapter settings option from the left navigation pane.t
5. Right-click the active network adapter and select the Properties option.
6. Select the Internet Protocol Version 4 (TCP/IPv4) option.
7. Click the Properties button.
8. Click the General tab.
9. Select the Use the following IP address option.
10. Specify a local IP address outside the local DHCP scope to prevent address conflicts – for example, 10.1.4.201.A quick tip: Usually, you'll find this information inside the DHCP settings section on the router. Since not every router is created equal, you may need to consult your manufacturer's support website for more specific details to find this information. If you're unsure of the configuration you have to use, use your current TCP/IP configuration as a reference, which you can review with the config /all command in Command Prompt.



11. Specify a subnet mask for the network (for example, 255.255.255.0), but the system usually fills this information from you based on the "IP address."
12. Specify the default gateway address, which should be the router's address — for example, 10.1.4.1.
13. Under the "Use the following DNS server addresses" section, in the "Preferred DNS server" field, specify the IP address of your DNS server, which in most cases is also the address of the router — for example, 10.1.4.1.A quick tip: If you can't connect to the internet, try using one of the Google Public DNS addresses (such as 8.8.8.8) for the "Alternate DNS server" option.
14. Click the OK button.
15. Click the Close button.