INDEX

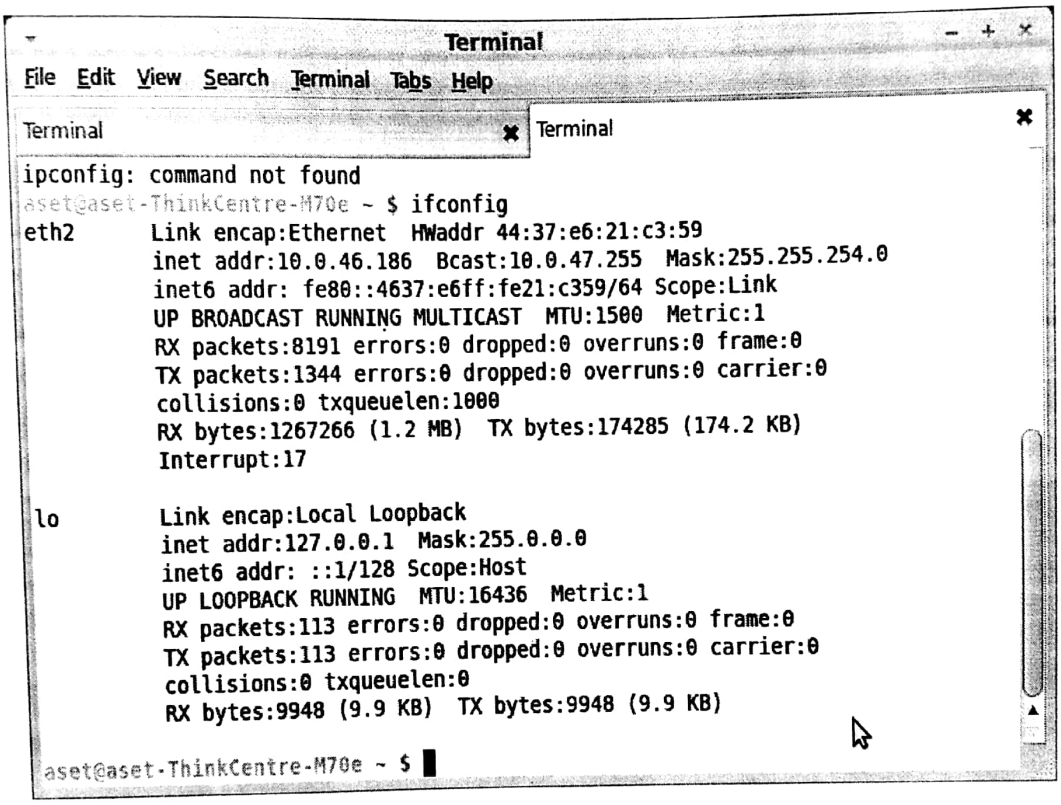
S.NO.	EXPERIMENT	DATE	SIGNATURE
1.	Network Commands on Linux / Unix	8/01/18	
2.	TO establish straight configuration for LAN	8/01/18	
3.	TO establish rollover conf. for LAN	8/01/18	05/02
4.	TO establish crossover conf. for LAN	8/01/18	
5.	TO translate dotted element of decimal IP add. into 32 bit add	15/01/18	
6.	TO determine if the IP add. is in Class A, B, C, D OR E	22/01/18	
7.	To find shortest path using ~~Bellman ford Algorithm~~ Dijkstra's Algo.	29/1/18	
8.	TO find shortest path using Bellman's Ford Algorithm	5/2/18	
9.	Program which manipulates Pure Aloha	12/2/18	26/3
10.	Program which manipulates Slotted ALOHA	9/2/18	
11.			

EXPERIMENT: 1

Linux commands

1. Ipconfig

It tells about IP Configuration Settings like IP Address etc.



2. Hostname

It is used to show or set a computer's hostname & domain name i.e. it displays system identity name. It is one of the most basic network administrative utility.

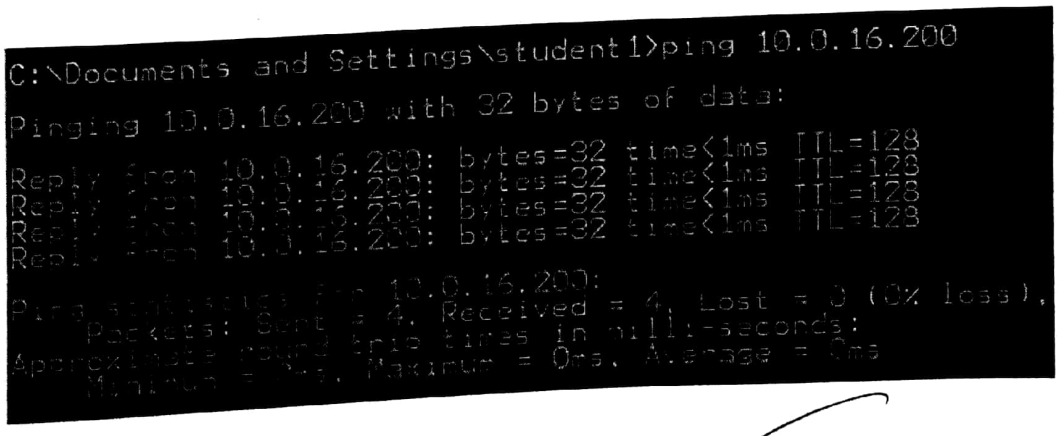
3. Ping

It is used to test the connection & latency between two network connections. These connections can be either in a Local Area Network or Wide Area Network or Internet as a whole.

Example-ping 10.0.16.200

Ping command has several options:-

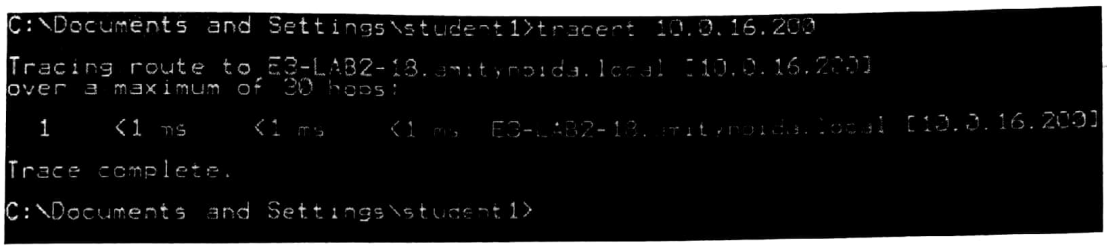
-t	Ping the specified host until stopped
-a	Resolves addresses to hostnames
-n count	No. of echo requests to send
-l size	Send buffer size
-f	Set Don't Fragment flag in packet
-l TTL	Time to Live
-v TOS	Type of service
-r count	Record route for count hops
-s count	Timestamp for count hops
-j host-list	Loose source route along host-list
-k host-list	Strict source route along host-list
-w timeout	Timeout in miliseconds to wait for each reply



4. Tracert

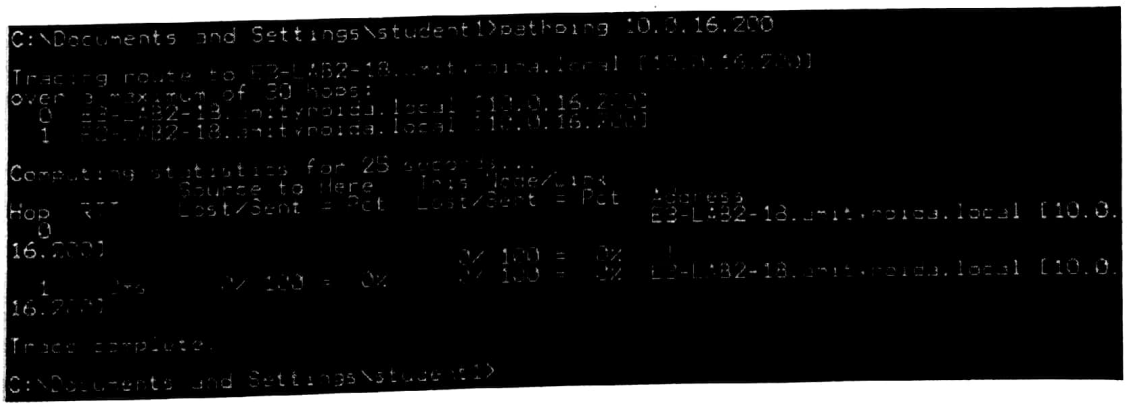
Trace route is a network tool used to determine the path packets taken from one IP Address to another.

Example- tracert 10.0.16.200



5. Pathping

It combines the functionality of ping with that of tracert by providing details of path between 2 hosts & ping like statistics for each node in the path based on samples taken over a period of time depending on how many are between the Start & End Host.



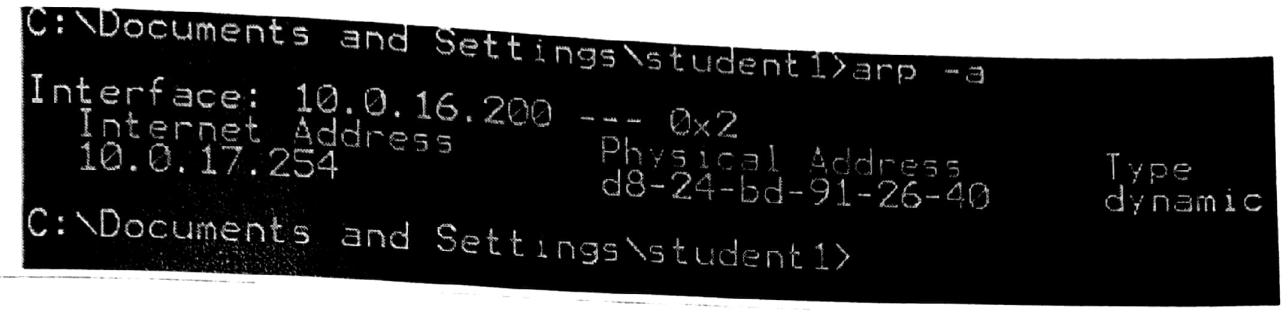
6. arp

It is used for address resolution. Its full form is Address Resolution Protocol.

arp is used to perform IP Address Resolution which is linking of IP Addresses to Mac Addresses.

Mac address is address of hardware device. MAC stands for Media Access Control.

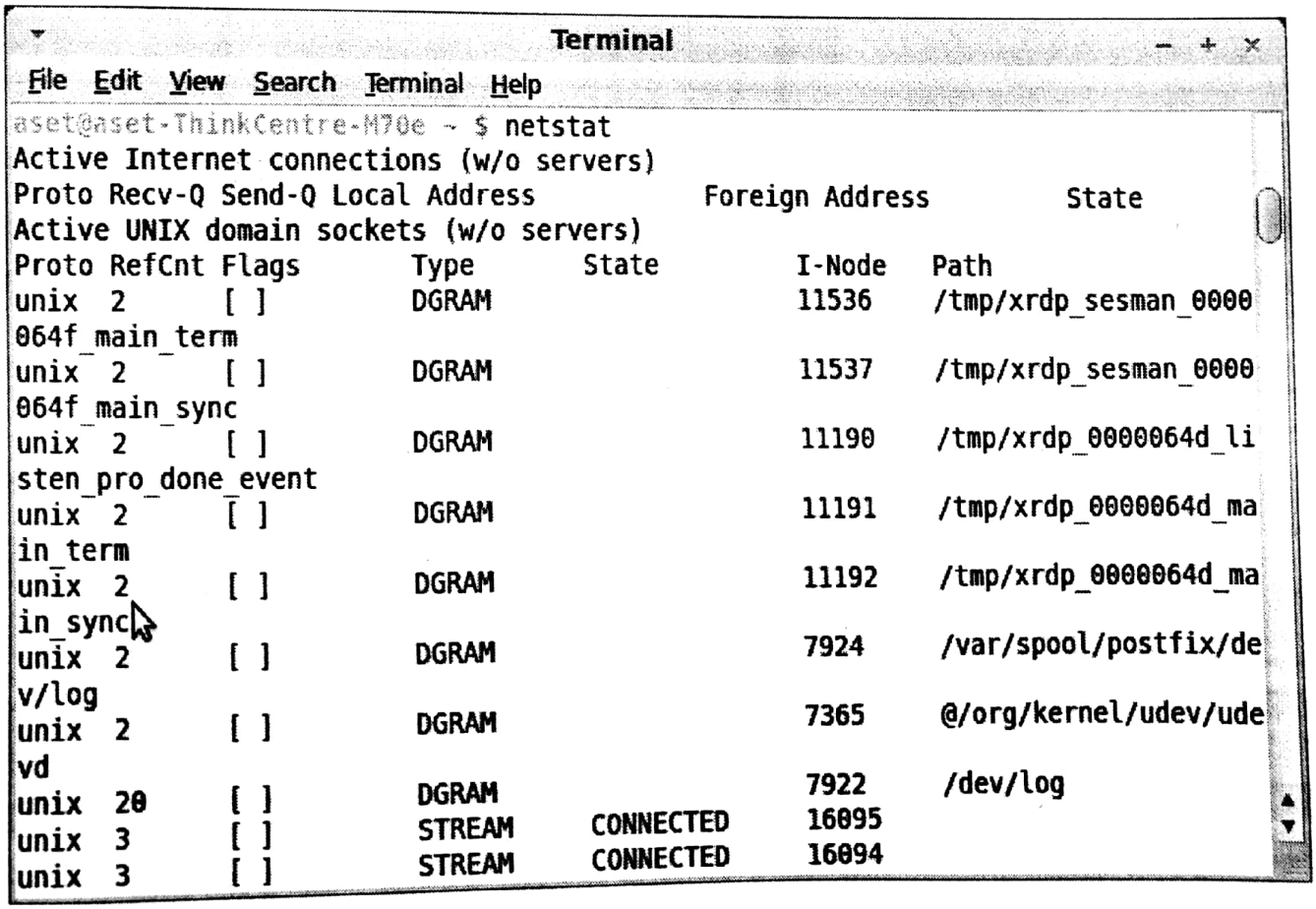
Arp uses a broadcast to do this by asking the host that has the given IP Address to the broadcast with its Mac Address.



7. Netstat

It is a command line tool that displays the network connections both incoming & outgoing, routing tables (Routing tables are on router which tell what are the routes/paths to send a packet & what delay would be there on which path) & a no. of network interface statistics.

It is used for finding problems in the network & to determine the amount of traffic on the network as a performance measurement.

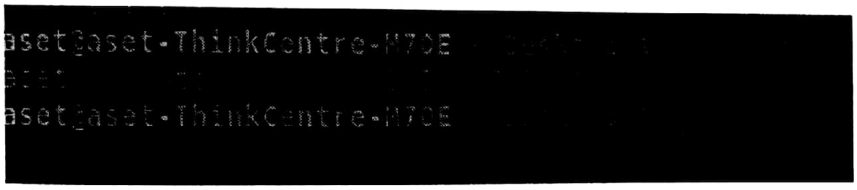


8. Finger

It is used to display information about a user on a specified system running the finger service. Output varies based on remote system. It works on Linux only.

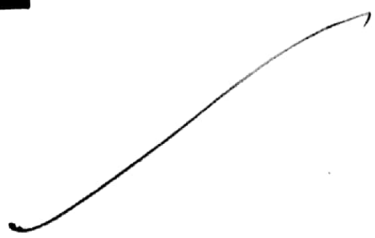
9. Who am i

This command is found on most Unix like Operating systems. It prints the effective user id, his terminal name and the date & time when the user logged on to the system.



10. Who

This command will tell the same data as who am I but it will tell for all the users logged in currently.



EXPERIMENT NO.-2

To establish straight configuration for LAN.

EQUIPMENTS REQUIRED:-

1) 2 RJ -45 connectors
2) Twisted pair cable
3) Gripping or Crimping tool
4) SLT-Tool

Colour Coding:-

1. Orange white
2. Orange
3. Green white
4. Blue
5. Blue white
6. Green
7. Brown white
8. Brown

Straight Configuration:-

1----------1

2----------2

3----------3

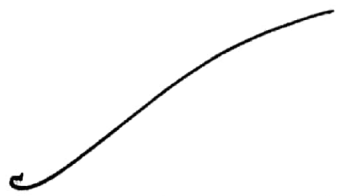
4----------4

5----------5

6----------6

7----------7

8----------8



THEORY: A **local area network (LAN)** is a computer network that connects computers and devices in a limited geographical area such as home, school, computer laboratory or office building. The defining characteristics of LANs, in contrast to wide area networks (WANs), include their usually higher data-transfer rates, smaller geographic area, and lack of a need for leased telecommunication lines.

ARCNET, Token Ring and other technology standards have been used in the past, but Ethernet over twisted pair cabling, and Wi-Fi are the two most common technologies currently in use.

PROCEDURE:-

The outer covering of the wire is peeled off and according to requirement the wires are inserted in RJ-45 connector and punched with the help of punching tool after punching the wire is tested with SLT(Side locator tool).

RESULT:-

The straight wiring for the LAN has been established and tested using SLT tool.

USE:-

This type of wiring is used for connecting to PC or Hub.

EXPERIMENT NO.-3

To establish rollover configuration for LAN.

EQUIPMENTS REQUIRED:-

1) 2 RJ -45 connectors
2) Twisted pair cable
3) Gripping or Crimping tool
4) SLT-Tool

Colour Coding:-

1. Orange white
2. Orange
3. Green white
4. Blue
5. Blue white
6. Green
7. Brown white
8. Brown

Rollover Configuration:-

1----------8

2----------7

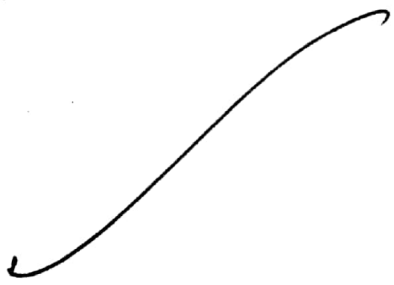
3----------6

4----------5

5----------4

6----------3

7----------2



THEORY: A **local area network (LAN)** is a computer network that connects computers and devices in a limited geographical area such as home, school, computer laboratory or office building. The defining characteristics of LANs, in contrast to wide area networks (WANs), include their usually higher data-transfer rates, smaller geographic area, and lack of a need for leased telecommunication lines.

ARCNET, Token Ring and other technology standards have been used in the past, but Ethernet over twisted pair cabling, and Wi-Fi are the two most common technologies currently in use.

PROCEDURE:-

The outer covering of the wire is peeled off and according to requirement the wires are inserted in RJ-45 connector and punched with the help of punching tool after punching the wire is tested with SLT(Side locator tool).

RESULT:-

The rollover wiring for the LAN has been established and tested using SLT tool.

USE:-

This type of wiring is used for configuration of the router.

EXPERIMENT NO.-4

To establish crossover configuration for LAN.

EQUIPMENTS REQUIRED:-

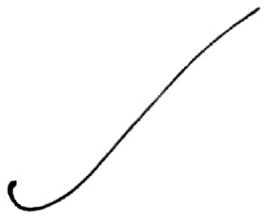
1) 2 RJ -45 connectors
2) Twisted pair cable
3) Gripping or Crimping tool
4) SLT-Tool

Colour Coding:-

1. Orange white
2. Orange
3. Green white
4. Blue
5. Blue white
6. Green
7. Brown white
8. Brown

Crossover Configuration:-

1----------3
2----------6
3----------1
4----------4
5----------5
6----------2
7----------7
8----------8



THEORY: A **local area network (LAN)** is a computer network that connects computers and devices in a limited geographical area such as home, school, computer laboratory or office building. The defining characteristics of LANs, in contrast to wide area networks (WANs), include their usually higher data-transfer rates, smaller geographic area, and lack of a need for leased telecommunication lines.

ARCNET, Token Ring and other technology standards have been used in the past, but Ethernet over twisted pair cabling, and Wi-Fi are the two most common technologies currently in use.

PROCEDURE:-

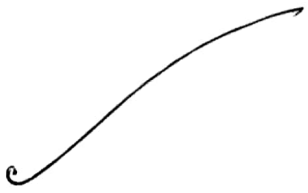
The outer covering of the wire is peeled off and according to requirement the wires are inserted in RJ-45 connector and punched with the help of punching tool after punching the wire is tested with SLT (Side locator tool).

RESULT:-

The crossover wiring for the LAN has been established and tested using SLT tool.

USES:

This type of wiring is used for connecting PC to another PC.



Experiment-05

OBJECTIVE: To translate dotted decimal IP address into 32 bit address.

SOFTWARE USED: Turbo C++.

THEORY: An Internet Protocol address (IP address) is a numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication. An IP address serves two principal functions: host or network interface identification and location addressing.

The designers of the Internet Protocol defined an IP address as a 32-bit number and this system, known as Internet Protocol Version 4 (IPv4), is still in use today. However, due to the enormous growth of the Internet and the predicted depletion of available addresses, a new addressing system (IPv6), using 128 bits for the address, was developed in 1995.

INPUT:

```
#include<iostream.h>
#include<conio.h>
void main()
        {
        clrscr();
        int i,j,a[4],bin[8]={128,64,32,16,8,4,2,1};
        cout<<"Enter the ip address";
        for(i=0;i<4;i++)
                cin>>a[i];
        cout<<"The ip address is:-"<<a[0]<<"."<<a[1]<<"."<<a[2]<<"."<<a[3]<<endl;
        for(i=0;i<4;i++)
                {
                for(j=0;j<8;j++)
                        {
                        if (a[i]&bin[j])
                        cout<<1;
                        else
```

```
                cout<<();
                }
        cout<<".";
        }
    getch();
    }
```

OUTPUT:

Experiment-06

OBJECTIVE: To determine if the IP address is in Class A, B, C, D, or E.

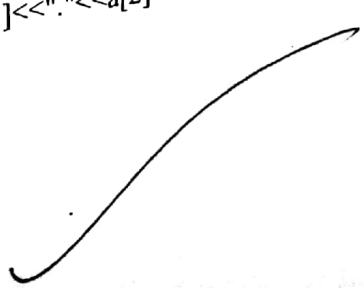
SOFTWARE USED: Turbo C++.

THEORY: An Internet Protocol address (IP address) is a numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication. An IP address serves two principal functions: host or network interface identification and location addressing. Its role has been characterized as follows: "*A name indicates what we seek. An address indicates where it is. A route indicates how to get there.*"

The designers of the Internet Protocol defined an IP address as a 32-bit number and this system, known as Internet Protocol Version 4 (IPv4), is still in use today. However, due to the enormous growth of the Internet and the predicted depletion of available addresses, a new addressing system (IPv6), using 128 bits for the address, was developed in 1995.

INPUT:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
    {
    clrscr();
    int a[4],i=0;
    cout<<"Enter The IP address";
    for(i=0;i<4;i++)
        cin>>a[i];
    cout<<"\n IP ADDRESS:"<<a[0]<<"."<<a[1]<<"."<<a[2]<<"."<<a[3]<<"\n";
    cout<<"The IP address is in Class: ";
    if(a[0]>=0 && a[0]<=127)
    cout<<"Class A";
    if(a[0]>127 && a[0]<191)
    cout<<"Class B";
```



```
if(a[0]>191 && a[0]<224)
cout<<"Class C";
if(a[0]>224 && a[0]<=239)
cout<<"Class D";
if(a[0]>239)
cout<<"Class E";
getch();
}
```

OUTPUT:

```
Enter The IP address
123
34
56
32

 IP ADDRESS:123.34.56.32
The IP address is in Class: Class A_
```

Experiment-07

OBJECTIVE: To implement the Dijkstra Algorithm to find shortest path.

SOFTWARE USED: Turbo C++.

THEORY: Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959, [1] [2] is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms.

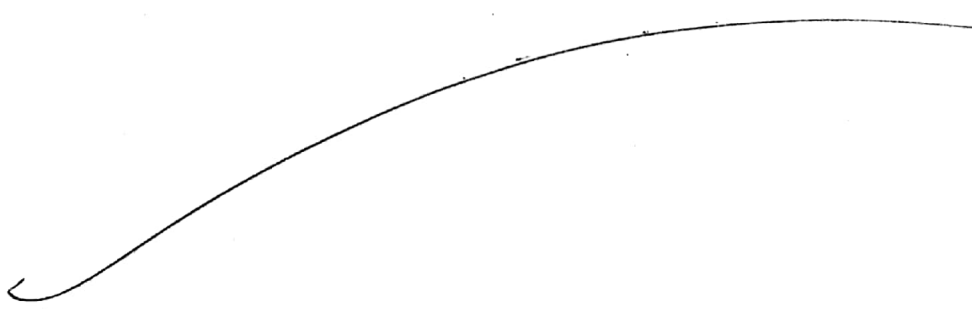
For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined.

For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path first is widely used in network routing protocols, most notably IS-IS and OSPF (Open Shortest Path First).

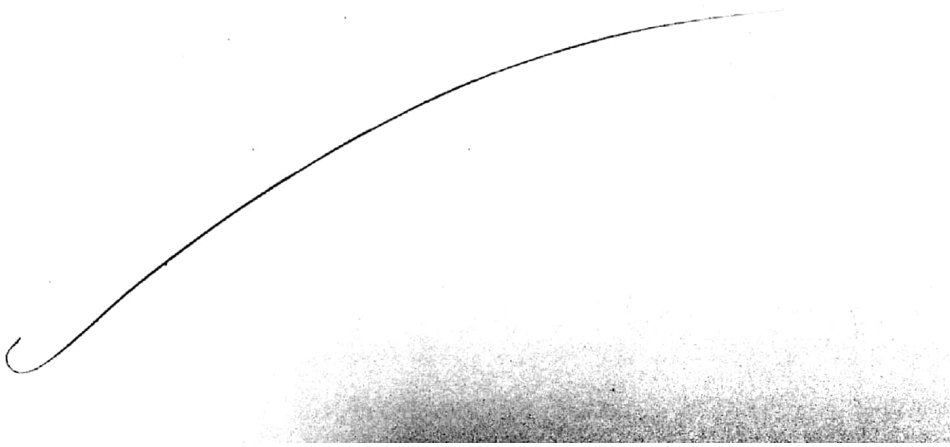
CODE:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
//using namespace std;
int shortest(int ,int);
int cost[10][10],dist[20],i,j,n,k,m,S[20],v,totcost,path[20],p;
main()
{
int c;
cout <<"enter no of vertices";
cin >> n;
cout <<"enter no of edges";
cin >>m;
```

```
cout <<"\nenter\nEDGE Cost\n";
for(k=1;k<=m;k++)
{
cin >> i >> j >>c;
cost[i][j]=c;
}
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
if(cost[i][j]==0)
cost[i][j]=31999;
cout <<"enter initial vertex";
cin >>v;
cout << v<<"\n";
shortest(v,n);
}

int shortest(int v,int n)
{
int min;
for(i=1;i<=n;i++)
{
S[i]=0;
dist[i]=cost[v][i];
}
path[++p]=v;
S[v]=1;
dist[v]=0;
for(i=2;i<=n-1;i++)
```

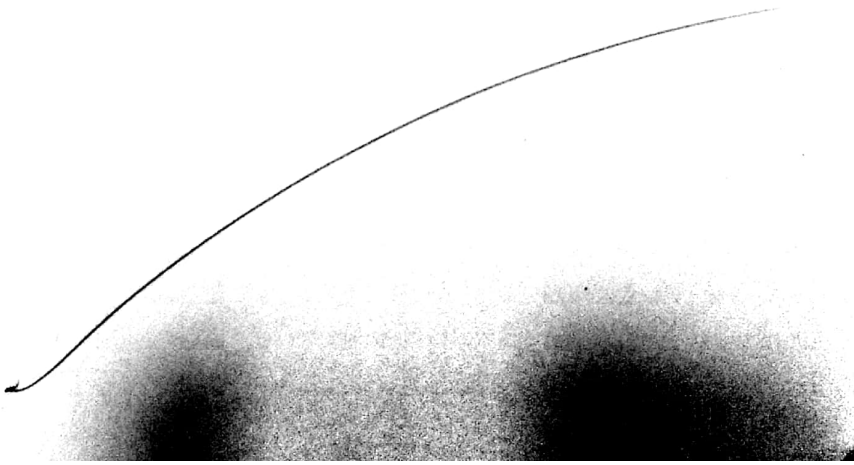


```
{
k=-1;
min=31999;
for(j=1;j<=n;j++)
{
if(dist[j]<min  && S[j]!=1)
{
min=dist[j];
k=j;
}
}
if(cost[v][k]<=dist[k])
p=1;
path[++p]=k;
for(j=1;j<=p;j++)
cout<<path[j];
cout <<"\n";
//cout <<k;
S[k]=1;
for(j=1;j<=n;j++)
if(cost[k][j]!=31999  && dist[j]>=dist[k]+cost[k][j]  && S[j]!=1)
 dist[j]=dist[k]+cost[k][j];
}
}
```



Output

```
enter no of vertices3
enter no of edges6

enter
EDGE Cost
1 2 25
1 3 35
1 4 10
2 3 10
2 4 15
3 5 30
enter initial vertex1
1
12
```



Experiment-8

OBJECTIVE: WAP to find shortest path using Bellman Ford's algorithm

SOFTWARE USED: Turbo C++.

THEORY: The **Bellman–Ford algorithm** is an algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph. The algorithm was first proposed by Alfonso Shimbel in 1955, but is instead named after Richard Bellman and Lester Ford, Jr., who published it in 1958 and 1956, respectively. Edward F. Moore also published the same algorithm in 1957, and for this reason it is also sometimes called the **Bellman–Ford–Moore algorithm**. Negative edge weights are found in various applications of graphs, hence the usefulness of this algorithm. If a graph contains a "negative cycle" (i.e. a cycle whose edges sum to a negative value) that is reachable from the source, then there is no *cheapest* path: any path that has a point on the negative cycle can be made cheaper by one more walk around the negative cycle. In such a case, the Bellman–Ford algorithm can detect negative cycles and report their existence.

CODE:

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#define INFINITY 999
struct node
{
   int cost;
   int value;
   int from;
}a[5];
void addEdge(int am[][5],int src,int dest,int cost)
{
    am[src][dest] = cost;
    return;
}
void bell(int am[][5])
{
   int i, j, k, c = 0, temp;
   a[0].cost = 0;
   a[0].from = 0;
   a[0].value = 0;
   for (i = 1; i < 5; i++)
   {
      a[i].from = 0;
      a[i].cost = INFINITY;
```

```
    a[i].value = 0;
}
while (c < 5)
{
    int min = 999;
    for (i = 0; i < 5; i++)
    {
        if (min > a[i].cost && a[i].value == 0)
        {
            min = a[i].cost;
        }
        else
        {
            continue;
        }
    }
    for (i = 0; i < 5; i++)
    {
        if (min == a[i].cost && a[i].value == 0)
        {
            break;
        }
        else
        {
            continue;
        }
    }
    temp = i;
    for (k = 0; k < 5; k++)
    {
        if (am[temp][k] + a[temp].cost < a[k].cost)
        {
            a[k].cost = am[temp][k] + a[temp].cost;
            a[k].from = temp;
        }
        else
        {
            continue;
        }
    }
    a[temp].value = 1;
    c++;
}
```

```
    cout<<"Cost"<<"\t"<<"Source Node"<<endl;
    for (j = 0; j < 5; j++)
    {
      cout<<a[j].cost<<"\t"<<a[j].from<<endl;
    }
}
int main()
{
  int n, am[5][5], c = 0, i, j, cost;
  for (i = 0; i < 5; i++)
  {
    for (j = 0; j < 5; j++)
    {
      am[i][j] = INFINITY;
    }
  }
  while (c < 8)
  {
    cout<<"Enter the source, destination and cost of edge\n";
    cin>>i>>j>>cost;
    addEdge(am, i, j, cost);
    c++;
  }
  bell(am);
  getch();
  return(0);
}
```

OUTPUT:

```
Enter the source, destination and cost of edge
0
1
-1
Enter the source, destination and cost of edge
0
2
4
Enter the source, destination and cost of edge
1
2
3
Enter the source, destination and cost of edge
1
3
2
Enter the source, destination and cost of edge
3
1
1
Enter the source, destination and cost of edge
1
4
2
```

```
Enter the source, destination and cost of edge
4
3
-3
Enter the source, destination and cost of edge
3
2
5
Cost    Source Node
0       0
-1      0
2       1
-2      4
1       1
```

Experiment-9

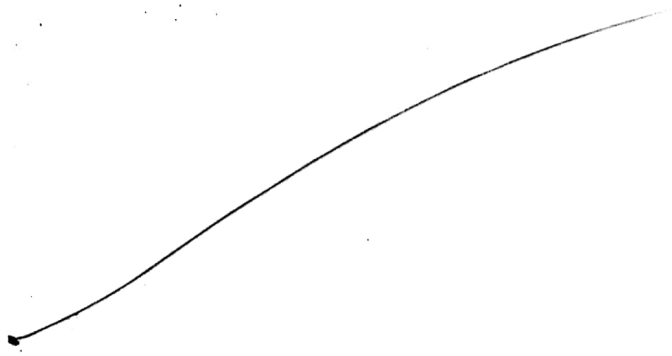
OBJECTIVE: Write a Program which manipulates pure ALOHA.

SOFTWARE USED: Turbo C++ , Windows OS

THEORY: A protocol for satellite and terrestrial radio transmissions. In pure Aloha, a user can transmit at any time but risks collisions with other users' messages. "Slotted Aloha" reduces the chance of collisions by dividing the channel into time slots and requiring that the user send only at the beginning of a time slot..

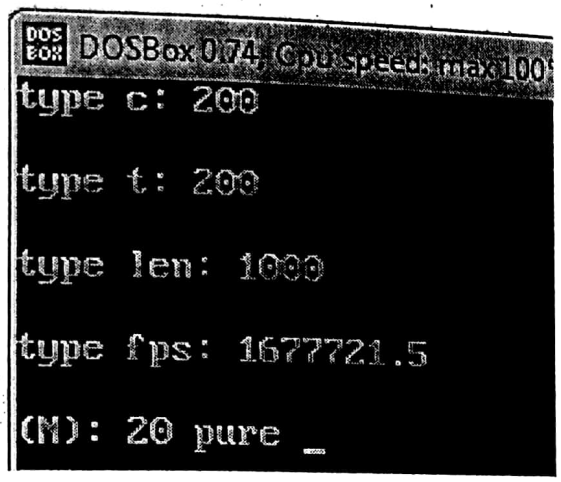
INPUT:

```
#include<iostream.h>

#include<conio.h>

int main()
{
clrscr();
int N=0;
float C=0,T=0,l=0,f=0,u=0,G=0;
cout<<"type c: ";a
cin>>C;

 cout<<"\ntype t: ";
 cin>>T;
 cout<<"\ntype len: ";
 cin>>l;
 cout<<"\ntype fps: ";
 cin>>f;

 C*=1024*1024*8;
 u=1/l;
 N=T*((C*u)-f);
 G=N/f;
 cout<<"\n(N): " <<N;

 if(G>0 && G<=0.5)
 cout<<" pure ";
 else if (G>0.5 && G<=1)
 cout<<" slotted";
 else
```



```
cout<<"\nerror.";
getch();
return 0;
}
```

OUTPUT:



Experiment-10

OBJECTIVE: Write a Program which manipulates slotted ALOHA.

SOFTWARE USED: Turbo C++ , Windows OS

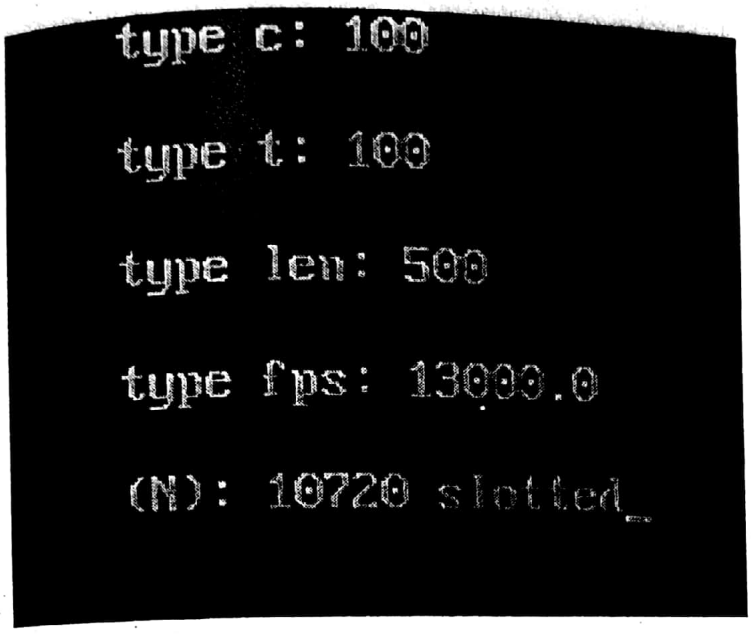
THEORY: Slotted Aloha reduces the chance of collisions by dividing the channel into time slots and requiring that the user send only at the beginning of a time slot. Aloha was the basis for Ethernet, a local area network protocol.

INPUT:

```
#include<iostream.h>

#include<conio.h>

int main()
{
clrscr();
int N=0;
float C=0,T=0,l=0,f=0,u=0,G=0;
cout<<"type c: ";a
cin>>C;

cout<<"\ntype t: ";
cin>>T;
cout<<"\ntype len: ";
cin>>l;
cout<<"\ntype fps: ";
cin>>f;

C*=1024*1024*8;
u=1/l;
N=T*((C*u)-f);
G=N/f;
cout<<"\n(N): " <<N;

if(G>0 && G<=0.5)
cout<<" pure ";
else if (G>0.5 && G<=1)
cout<<" slotted";
else
cout<<"\nerror.";
```

```
getch();
return 0;
}
```

OUTPUT:



26/03