EXPERIMENT NO 1

IPCONFIG

```
C:\Users\Student>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.16.51.69
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . : 172.16.48.2
```

IPCONFIG/ALL

```
C:\Users\Student>ipconfig/all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : sclab35
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Ethernet Connection (14) I219-V
   Physical Address. . . . . . . . . : A8-A1-59-DA-DB-99
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 172.16.51.69(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Lease Obtained. . . . . . . . . . : 08 February 2023 18:24:06
   Lease Expires . . . . . . . . . . : 09 February 2023 18:24:07
   Default Gateway . . . . . . . . . : 172.16.48.2
   DHCP Server . . . . . . . . . . . : 172.16.48.2
   DNS Servers . . . . . . . . . . . : 8.8.8.8
                                       218.248.255.139
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

PING

```
C:\Users\Student>ping google.com

Pinging google.com [142.250.193.174] with 32 bytes of data:
Reply from 142.250.193.174: bytes=32 time=18ms TTL=59
Reply from 142.250.193.174: bytes=32 time=18ms TTL=59
Reply from 142.250.193.174: bytes=32 time=18ms TTL=59
Reply from 142.250.193.174: bytes=32 time=18ms TTL=59

Ping statistics for 142.250.193.174:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 18ms, Maximum = 18ms, Average = 18ms
```

ARP -A

```
C:\Users\Student>arp -a

Interface: 172.16.51.69 --- 0xe
  Internet Address      Physical Address      Type
  172.16.48.2           7c-5a-1c-cf-50-b9     dynamic
  172.16.49.191         d0-67-e5-15-c9-7e     dynamic
  172.16.49.192         d0-67-e5-15-c9-c3     dynamic
  172.16.49.193         d0-67-e5-16-48-61     dynamic
  172.16.49.194         d0-67-e5-15-ca-8c     dynamic
  172.16.49.195         e0-d5-5e-d9-92-7e     dynamic
  172.16.49.197         d0-67-e5-15-ca-c1     dynamic
  172.16.49.199         d0-67-e5-15-cc-b2     dynamic
  172.16.49.202         d0-67-e5-16-63-13     dynamic
  172.16.49.204         10-78-d2-55-15-06     dynamic
  172.16.49.216         d0-67-e5-16-48-b3     dynamic
```

TRACERT

```
C:\Users\Student>tracert google.com

Tracing route to google.com [142.250.76.78]
over a maximum of 30 hops:

  1    12 ms    <1 ms    <1 ms  172.16.48.2
  2    50 ms    51 ms    48 ms  172.24.71.66
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5    32 ms    19 ms    20 ms  72.14.218.250
  6    19 ms    19 ms    19 ms  142.251.227.211
  7    20 ms    20 ms    20 ms  142.250.228.245
  8    20 ms    20 ms    21 ms  maa05s14-in-f14.1e100.net [142.250.76.78]

Trace complete.
```

PATHPING

```
C:\Users\Student>pathping google.com

Tracing route to google.com [142.250.76.78]
over a maximum of 30 hops:
  0  sclab35 [172.16.51.69]
  1  172.16.48.2
  2  172.24.71.66
  3  117.216.207.223
  4     *        *        *
Computing statistics for 75 seconds...
            Source to Here   This Node/Link
Hop  RTT    Lost/Sent = Pct  Lost/Sent = Pct  Address
  0                                           sclab35 [172.16.51.69]
                                 0/ 100 =  0%   |
  1    0ms     0/ 100 =  0%     0/ 100 =  0%  172.16.48.2
                                 0/ 100 =  0%   |
  2    3ms     0/ 100 =  0%     0/ 100 =  0%  172.24.71.66
                                 0/ 100 =  0%   |
  3   21ms     0/ 100 =  0%     0/ 100 =  0%  117.216.207.223

Trace complete.
```

GETMAC

```
C:\Users\Student>getmac

Physical Address    Transport Name
=================== ============================================================
A8-A1-59-DA-DB-99   \Device\Tcpip_{F490564B-8977-4654-A7B5-E5E237E090B1}
```

SYSTEMINFO

```
C:\Users\Student>systeminfo

Host Name:                 sclab35
OS Name:                   Microsoft Windows 11 Pro
OS Version:                10.0.22000 N/A Build 22000
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          SC-LAB
Registered Organization:
Product ID:                00331-20463-51822-AA972
Original Install Date:     04-07-2022, 23:15:58
System Boot Time:          06-02-2023, 19:17:00
System Manufacturer:       Acer
System Model:              Veriton S2680G
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 167 Stepping 1 GenuineIntel ~2592 Mhz
BIOS Version:              American Megatrends International, LLC. P1.40L, 17-01-2022
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi
Total Physical Memory:     15,975 MB
Available Physical Memory: 9,805 MB
Virtual Memory: Max Size:  18,407 MB
Virtual Memory: Available: 12,206 MB
Virtual Memory: In Use:    6,201 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              \\sclab35
Hotfix(s):                 4 Hotfix(s) Installed.
                           [01]: KB5022406
                           [02]: KB5012170
                           [03]: KB5022287
                           [04]: KB5019385
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) Ethernet Connection (14) I219-V
                                 Connection Name: Ethernet
                                 DHCP Enabled:    Yes
                                 DHCP Server:     172.16.48.2
                                 IP address(es)
                                 [01]: 172.16.51.69
Hyper-V Requirements:      VM Monitor Mode Extensions: Yes
                           Virtualization Enabled In Firmware: Yes
                           Second Level Address Translation: Yes
                           Data Execution Prevention Available: Yes
```

NETSTAT

```
C:\Users\Student>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:53351        sclab35:53352          ESTABLISHED
  TCP    127.0.0.1:53352        sclab35:53351          ESTABLISHED
  TCP    127.0.0.1:53353        sclab35:53354          ESTABLISHED
  TCP    127.0.0.1:53354        sclab35:53353          ESTABLISHED
  TCP    172.16.51.69:7680      172.16.50.17:52710     TIME_WAIT
  TCP    172.16.51.69:7680      172.16.51.70:51823     TIME_WAIT
```

NETSTAT -R

```
C:\Users\Student>netstat -r
===========================================================================
Interface List
 14...a8 a1 59 da db 99 ......Intel(R) Ethernet Connection (14) I219-V
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      172.16.48.2     172.16.51.69     25
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      172.16.48.0    255.255.240.0         On-link      172.16.51.69    281
     172.16.51.69  255.255.255.255         On-link      172.16.51.69    281
    172.16.63.255  255.255.255.255         On-link      172.16.51.69    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link      172.16.51.69    281
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link      172.16.51.69    281
===========================================================================
Persistent Routes:
  None

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
  1    331 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```

ROUTE PRINT -4

```
C:\Users\Student>route print -4
===========================================================================
Interface List
 14...a8 a1 59 da db 99 ......Intel(R) Ethernet Connection (14) I219-V
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      172.16.48.2     172.16.51.69     25
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      172.16.48.0    255.255.240.0         On-link      172.16.51.69    281
     172.16.51.69  255.255.255.255         On-link      172.16.51.69    281
    172.16.63.255  255.255.255.255         On-link      172.16.51.69    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link      172.16.51.69    281
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link      172.16.51.69    281
===========================================================================
Persistent Routes:
  None
```

ROUTE PRINT -6

```
C:\Users\Student>route print -6
===========================================================================
Interface List
 14...a8 a1 59 da db 99 ......Intel(R) Ethernet Connection (14) I219-V
  1...........................Software Loopback Interface 1
===========================================================================

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
  1    331 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```

NSLOOKUP

```
C:\Users\Student>nslookup google.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    google.com
Addresses:  2404:6800:4007:815::200e
          142.250.182.142
```

EXPERIMENT NO 2

PROGRAM

```
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#include<sys/types.h>
int main(){
   int pid,ppid,i,num1,num2;
   if(fork()==0){
      pid=getpid();
      ppid=getppid();
      printf("Child process starts\n");
      printf("Process id of child process= %d\n",pid);
      printf("Process id of parent process from child process = %d\n",ppid);
      printf("Enter any two numbers :- ");
      scanf("%d%d",&num1,&num2);
      printf("sum = %d\n",num1+num2);
      printf("Child process end \n");

   }
   else{
      pid=getpid();
      printf("Parent process starts\n");
      printf("Process id of parent process= %d\n",pid);
      wait(NULL);
      printf("Ten numbers are :- \n");
      for(i=1;i<=10;i++){
            printf("%d\n",i);
      }
      printf("Parent process end \n");
   }
   return 0;
}
```

OUTPUT

```
Parent process starts
Child process starts
Process id of parent process= 35188
Process id of child process= 35189
Process id of parent process from child process = 35188
Enter any two numbers :- 8
4
sum = 12
Child process end
Ten numbers are :-
1
2
3
4
5
6
7
8
9
10
Parent process end
```

EXPERIMENT NO 3

PROGRAM

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main(){
    int num1,num2,pid;
    pid=getpid();
    printf("Proces id from program 1:- %d\n",pid);
    printf("Enter any two numbers :- ");
    scanf("%d%d",&num1,&num2);
    printf("sum = %d\n",num1+num2);
    char *args[]={"4","Hello",NULL};
    execv("./pg2",args);
    return 0;
}
```

PRORAM

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    int i,pid;
    pid=getpid();
    printf("Proces id from program 2:- %d\n",pid);
    for(i=0;i<argc;i++){
       printf("Argument %d :- %s\n",i,argv[i]);
    }
    return 0;
}
```

OUTPUT

```
Proces id from program 1:- 35943
Enter any two numbers :- 4
5
sum = 9
Proces id from program 2:- 35943
Argument 0 :- 4
Argument 1 :- Hello
```

EXPERIMENT NO 4

PROGRAM

Server.c

```
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <unistd.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr
void func(int connfd){
	char buff[MAX];
	int n;
	// infinite loop for chat
	while(1) {
	  bzero(buff, MAX);
	  read(connfd, buff, sizeof(buff));
	  printf("From client: %s\t To client : ", buff);
	  bzero(buff, MAX);
	  n = 0;
	  while ((buff[n++] = getchar()) != '\n');
	  write(connfd, buff, sizeof(buff));
	  if (strncmp("exit", buff, 4) == 0) {
		printf("Server Exit...\n");
		break;
	  }
	}
}
int main()
{
	int sockfd, connfd, len;
	struct sockaddr_in servaddr, cli;
	sockfd = socket(AF_INET, SOCK_STREAM, 0);
	if (sockfd == -1) {
	  printf("socket creation failed...\n");
	  exit(0);
	}
	else
	  printf("Socket successfully created..\n");
	bzero(&servaddr, sizeof(servaddr));
	servaddr.sin_family = AF_INET;
	servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
	servaddr.sin_port = htons(PORT);
	if ((bind(sockfd, (SA*)&servaddr, sizeof(servaddr))) != 0) {
	  printf("socket bind failed...\n");
	  exit(0);
	}
	else
	  printf("Socket successfully binded..\n");
	if ((listen(sockfd, 5)) != 0) {
	  printf("Listen failed...\n");
	  exit(0);
```

```
      }
      else
        printf("Server listening..\n");
      len = sizeof(cli);
      connfd = accept(sockfd, (SA*)&cli, &len);
      if (connfd < 0) {
        printf("server accept failed...\n");
        exit(0);
      }
      else
        printf("server accept the client...\n");
      func(connfd);
      close(sockfd);
}
```

Client.c

```
#include <arpa/inet.h> // inet_addr()
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h> // bzero()
#include <sys/socket.h>
#include <unistd.h> // read(), write(), close()
#define MAX 80
#define PORT 8080
#define SA struct sockaddr
void func(int sockfd)
{
    char buff[MAX];
    int n;
    for (;;) {
       bzero(buff, sizeof(buff));
       printf("Enter the string : ");
       n = 0;
       while ((buff[n++] = getchar()) != '\n')
             ;
       write(sockfd, buff, sizeof(buff));
       bzero(buff, sizeof(buff));
       read(sockfd, buff, sizeof(buff));
       printf("From Server : %s", buff);
       if ((strncmp(buff, "exit", 4)) == 0) {
             printf("Client Exit...\n");
             break;
       }
    }
}
int main(){
    int sockfd, connfd;
    struct sockaddr_in servaddr, cli;
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1) {
       printf("socket creation failed...\n");
       exit(0);
    }
    else
       printf("Socket successfully created..\n");
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
```

```
    servaddr.sin_addr.s_addr = inet_addr("172.16.112.99");
    servaddr.sin_port = htons(PORT);
    if (connect(sockfd, (SA*)&servaddr, sizeof(servaddr))!= 0){
       printf("connection with the server failed...\n");
       exit(0);
    }
    else
       printf("connected to the server..\n");

    func(sockfd);
    close(sockfd);
}
```

OUTPUT

Server

```
Socket successfully created..
Socket successfully binded..
Server listening..
server accept the client...
From client: Hii
         To client : Hello
From client: Okk
         To client : Bye
```

Client

```
Socket successfully created..
connected to the server..
Enter the string : Hii
From Server : Hello
Enter the string : Okk
From Server : Bye
Enter the string :
```

EXPERIMENT NO 5

PROGRAM

server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main(){
 char *ip = "127.0.0.1";
 int port = 8080;
 int sockfd;
 struct sockaddr_in server_addr, client_addr;
 char buffer[1024];
 char str[1024];
 socklen_t addr_size;
 int n;
 sockfd = socket(AF_INET, SOCK_DGRAM, 0);
 if (sockfd < 0) {
   perror("[-]socket error");
   exit(1);
 }
 memset(&server_addr, '\0', sizeof(server_addr));
 server_addr.sin_family = AF_INET;
 server_addr.sin_port = htons(port);
 server_addr.sin_addr.s_addr = inet_addr(ip);
 n = bind(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
 if (n < 0){
   perror("[-]bind error");
   exit(1);
 }
printf("Server created successfully !\n");
while(1){
 bzero(buffer, 1024);
 addr_size = sizeof(client_addr);
 recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr,
&addr_size);
 printf("[+]Data recv: %s\n", buffer);
 if(strcmp(buffer,"exit\n")==0){
   exit(0);
 }
 printf("Enter a message :-");
 bzero(buffer, 1024);
 n=0;
 while ((buffer[n++] = getchar()) != '\n');
 sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr,
sizeof(client_addr));
 printf("[+]Data send: %s\n", buffer);
 if(strcmp(buffer,"exit\n")==0){
   exit(0);
 }
}
 return 0;
}
```

client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main(){
  char *ip = "127.0.0.1";
  int port = 8080;
  int n;
  int sockfd;
  struct sockaddr_in addr;
  char buffer[1024];
  char str[1024];
  socklen_t addr_size;

  sockfd = socket(AF_INET, SOCK_DGRAM, 0);
  memset(&addr, '\0', sizeof(addr));
  addr.sin_family = AF_INET;
  addr.sin_port = htons(port);
  addr.sin_addr.s_addr = inet_addr(ip);

  printf("Client created successfully !\n");
while(1){
  printf("Enter a message :-");
  bzero(buffer, 1024);
  n=0;
  while ((buffer[n++] = getchar()) != '\n');
  sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&addr, sizeof(addr));
  printf("[+]Data send: %s\n", buffer);
  if(strcmp(buffer,"exit\n")==0){
      exit(0);
  }
  bzero(buffer, 1024);
  addr_size = sizeof(addr);
  recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr*)&addr, &addr_size);
  printf("[+]Data recv: %s\n", buffer);
  if(strcmp(buffer,"exit\n")==0){
      exit(0);
  }
}
  return 0;
}
```

OUTPUT

Server

```
Server created successfully !
[+]Data recv: Hii

Enter a message :-Hello
[+]Data send: Hello

[+]Data recv: How r u

Enter a message :-Fine
[+]Data send: Fine

[+]Data recv: exit
```

Client

```
Client created successfully !
Enter a message :-Hii
[+]Data send: Hii

[+]Data recv: Hello

Enter a message :-How r u
[+]Data send: How r u

[+]Data recv: Fine

Enter a message :-exit
[+]Data send: exit
```

EXPERIMENT NO 6

PROGRAM

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
struct frame{
      int info;
      int seq;
};
int ak;
int t = 5, k;
int disconnect = 0;
struct frame p;
char turn = 's'; // Initialize first turn as sender
int errorframe = 1; // no Error
int errorack = 1;
void sender();
void receiver();
void main(){
      p.info = 0; // data part
      p.seq = 0; // sequence number
      while (!disconnect){
      sender(); // call sender
      sleep(5);
      // After a finite amount of time call receiver
      receiver();
      }
}
void sender(){
      static int flag = 0;
      if (turn == 's') // sender turn{
      if (errorack == 0){ // Ack didn't arrive{
            printf("SENDER: sent packet with seq NO:%d\n", p.seq);
            errorframe = rand() % 4; // randomly pick Error frame as 4
            printf("%s\n", (errorframe == 0 ? "Error While sending Packet" :
""));
            turn = 'r';
      }
      else{
            if (flag == 1)
            printf("SENDER: Received ACK for packet %d\n", ak);
            if (p.seq == 5){
            disconnect = 1;
            return;
            }
            p.info = p.info + 1;
            p.seq = p.seq + 1;
            printf("SENDER: sent packet with seq NO:%d\n", p.seq);
            errorframe = rand() % 4;
            // Message below is printed only if Error
            // occured while sending Packet
            printf("%s\n", (errorframe == 0 ? "Error While sending Packet" :
""));
            turn = 'r';
            // Set next turn as Receiver for transmission
            flag = 1;
      }
      else{
```

```
	t--;
	printf("SENDER time reducing\n");
	if (t == 0){
		turn = 's';
		errorack = 0;
		t = 5;
	}
	}
}
void receiver(){
	static int frexp = 1;
	if (turn == 'r'){
	if (errorframe != 0)
	{
		if (p.seq == frexp){
		// if frame sequence number is eq to frexp
		printf("RECEIVER: Received packet with seq %d\n", p.seq);
		// note sequence number of frame arrived
		// to send acknowledgement
		ak = p.seq;
		// increment the frame sequence number
		frexp = frexp + 1;
		// Set next turn as sender
		turn = 's';
		// Send acknowledgement error for frame number 4
		errorack = rand() % 4;
		printf("%s\n", (errorack == 0 ? "Error While sending ACK" :
""));
		}
		else{
		// Receiver received Duplicated frame for lost frame after
Resending
		printf("RECEIVER: Duplicated packet with seq %d\n", frexp - 1);
		// Note down acknowledgement number of frame
		ak = frexp - 1;
		// next turn sender
		turn = 's';
		errorack = rand() % 4;
		printf("%s\n", (errorack == 0 ? "Error While sending ACK" :
""));
		}
	}
	}
}
```

OUTPUT

```
SENDER: sent packet with seq NO:1

RECEIVER: Received packet with seq 1

SENDER: Received ACK for packet 1
SENDER: sent packet with seq NO:2

RECEIVER: Received packet with seq 2

SENDER: Received ACK for packet 2
SENDER: sent packet with seq NO:3

RECEIVER: Received packet with seq 3

SENDER: Received ACK for packet 3
SENDER: sent packet with seq NO:4

RECEIVER: Received packet with seq 4
Error While sending ACK
SENDER: sent packet with seq NO:4

RECEIVER: Duplicated packet with seq 4

SENDER: Received ACK for packet 4
SENDER: sent packet with seq NO:5

RECEIVER: Received packet with seq 5

SENDER: Received ACK for packet 5
```

EXPERIMENT NO 7

PROGRAM

```
#include<stdio.h>
#include <unistd.h>
int main(){
      int buffS,buffSize,noPackets,i,bucket[50],temp,count=0,packetS[50],
rateF,time,dataSent=0;
      printf("Enter the buffer size :- ");
      scanf("%d",&buffSize);
      buffS=buffSize;
      printf("Enter the no. of packets :-");
      scanf("%d",&noPackets);
      for(i=0;i<noPackets;i++){
      printf("Size of Packet [%d] :- ",i);
      scanf("%d",&temp);
      if(temp>buffS){
            printf("Packet size is grater then buffer space\n");
      }
      else{
            bucket[count]=i;
            buffS-=temp;
            packetS[count]=temp;
            count++;
      }
      }
      printf("\nEnter the rate of data flow :- ");
      scanf("%d",&rateF);
      printf("Time interval of data flow :- ");
      scanf("%d",&time);
      for(i=0;i<count;i++){
      while(packetS[i]>0){
            sleep(time);
            packetS[i]-=rateF;
            dataSent++;
      }
      printf("Packet[%d] completed in %d send\n",bucket[i],dataSent);
      }
}
```

OUTPUT

```
Enter the buffer size :- 50
Enter the no. of packets :-5
Size of Packet [0] :- 25
Size of Packet [1] :- 40
Packet size is grater then buffer space
Size of Packet [2] :- 10
Size of Packet [3] :- 5
Size of Packet [4] :- 2

Enter the rate of data flow :- 2
Time interval of data flow :- 1
Packet[0] completed in 13 send
Packet[2] completed in 18 send
Packet[3] completed in 21 send
Packet[4] completed in 22 send
```

EXPERIMENT NO 8

PROGRAM

```
#include <stdio.h>
#include <limits.h>
#define MAX_NODES 10
struct nodes {
      int dist[MAX_NODES];
      int through[MAX_NODES];
};
int main() {
      int n, i, j, k, matrix[MAX_NODES][MAX_NODES],
distanceVector[MAX_NODES][MAX_NODES], through[MAX_NODES][MAX_NODES];
      printf("Enter number of nodes: ");
      scanf("%d", &n);
      struct nodes node[n];
      printf("Enter the distance between each node (if there is no
connection, input -1):\n");
      for (i = 0; i < n; i++) {
      for (j = 0; j < n; j++) {
            if (i != j) {
            printf("Distance between %d and %d: ", i, j);
            scanf("%d", &node[i].dist[j]);
            if (node[i].dist[j] != -1) {
                  node[i].through[j] = j;
            } else {
                  node[i].through[j] = -1;
            }
            } else {
            node[i].dist[j] = 0;
            node[i].through[j] = -1;
            }
      }
      }
      printf("\nInitial distance table:\n");
      printf("\t");
      for(i=0;i<n;i++){
      printf("%d\t",i);
      }
      printf("\n");
      for (i = 0; i < n; i++) {
      printf("Node %d:\t", i);
      for (j = 0; j < n; j++) {
            printf("%d\t", node[i].dist[j]);
            node[i].through[j]=j;
      }
      printf("\n");
      }
      for (k = 0; k < n; k++) {
      for (i = 0; i < n; i++) {
            for (j = 0; j < n; j++) {
            if (node[i].dist[k] != -1 && node[k].dist[j] != -1) {
                  int newDistance = node[i].dist[k] + node[k].dist[j];
                  if (newDistance < node[i].dist[j] || node[i].dist[j] ==
-1) {
                        node[i].dist[j] = newDistance;
                        node[i].through[j] = node[i].through[k];
                  }
```

```
                }
                }
        }
        }
        for (i = 0; i < n; i++) {
        printf("\nDistance vector table for node %d:\n", i);
        printf("Node\tDistance   Through\n");
        for(j=0;j<n;j++){
                printf(" %d\t",j);
                printf("   %d\t", node[i].dist[j]);
                printf("\t%d\n", node[i].through[j]);
        }
        }
        return 0;
}
```

OUTPUT

```
Enter number of nodes: 3
Enter the distance between each node (if there is no connection, input -1):
Distance between 0 and 1: 2
Distance between 0 and 2: -1
Distance between 1 and 0: 2
Distance between 1 and 2: 3
Distance between 2 and 0: -1
Distance between 2 and 1: 3

Initial distance table:
        0       1       2
Node 0: 0       2       -1
Node 1: 2       0       3
Node 2: -1      3       0

Distance vector table for node 0:
Node    Distance     Through
 0         0            0
 1         2            1
 2         5            1

Distance vector table for node 1:
Node    Distance     Through
 0         2            0
 1         0            1
 2         3            2

Distance vector table for node 2:
Node    Distance     Through
 0         5            1
 1         3            1
 2         0            2
```