Output:

```
$ ./sender
 ^C
$
```

sender is executed,
after a while it is aborted.

```
$ ./listener
   RVCE-CSE
   RVCE-CSE
   RVCE-CSE
   ^C
$
```

The listener is started,
allowed to receive messages
for a while & then
stopped.
The messages are received
with a 1s interval.

```
$./listener
  RVCE-CSE
  RVCE-CSE
  RVCE-CSE
  RVCE-CSE
  RVCE-CSE
  RVCE-CSE
  RVCE-CSE
  ^C
$
```

The second listener is allowed
to run simultaneously.

It too receives messages
from the sender at
an interval of 1s.

4. Implement a simple muticast routing mechanism.

```
/* Listener */

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

#define  HELLO-PORT  12345
#define  HELLO-GROUP  "255.0.0.37"
#define  MSGBUFSIZE  25

int main ( )
{
        struct sockaddr_in addr;
        //intfd
        int fd , nbytes , addrlen ;
        struct ip_mreq mreq ;
        char msgbuf [MSGBUFSIZE];
        uint yes = 1;
        if((fd = socket ( AF INET, SOCK-DGRAM , 0))<0) {
                        printf(" Socket error");
                           exit(0);      }       //cont..
```

R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date ________ Name M.C.SOHAN

Dept./Lab ________ Class ________ Expt./No. 4

Title listener - multicast.

```
// modify the original socket.
if ( setsockopt ( fd, SOL_SOCKET , SO_REUSEADDR , &yes,
                sizeof (yes) ) < 0)
{
      printf ("Error"); exit (0); }

// set the destination address
memset (&addr, 0, sizeof (addr));
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl ( INADDR_ANY);
addr.sin_port = htons ( HELLO_PORT);
if (bind( fd, (struct sockaddr*) &addr, sizeof(addr))
          < 0)
  { printf ("Bind error"); exit (0); }
mreq.imr_multiaddr.s_addr = inet_addr (HELLO_GROUP);
mreq.imr_interface.s_addr = htonl ( INADDR_ANY);
```

Signature of
Teacher incharge

```
if ( setsockopt ( fd , IPPROTO_IP , IP_ADD_MEMBERSH
                    & mreq , sizeof (mreq)) <0 )
{     printf ("sockopt error"); exit(0); }

   // read - print loop.

while (1)
{
      addrlen = sizeof (addr);
      if ( (nbytes = recvfrom( fd, msgbuf, MSGBU
                   ,0, (struct sockaddr *)&addr ,
                      &addrlen ))<0)
            printf ("Error in recvfrom");
        puts (msgbuf);
}
return 0;
} // end of listener.
```

pg.10

R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date ______________ Name M.C,SOHAN.

Dept./Lab ______________ Class ______________ Expt./No. 4

Title Sender

```
// sender-multicast.

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

#define HELLO_PORT  12345
#define HELLO-GROUP "225.0.0.37"

int main()
{
        struct sockaddr_in addr;
        int fd, cnt;
        struct ip-mreq;
        Char *msg = "RVCE-CSE";
```

Signature of
Teacher incharge

```
if ( (fd = socket( AF_INET, SOCK_DGRAM, 0 ) ) <0 ) {
        printf (" socket error");
        exit(0);
  }

memset( &addr, 0, sizeof(addr));
 addr.sin_family = AF_INET;
 addr.sin-addr.saddr = inet-addr( HELLO_GROU
 addr.sin_port = htons( HELLO-PORT);

while(1)
{
     if( sendto( fd, msg, sizeof (msg) , 0,
              (struct sockaddr *) &addr, sizeof(add
              <0 ) {
        printf (" send error"); exit(0);}

       sleep(1);

  }
  return 0;

} //end of program.
```

xpt. No. 4

//end.