SERVER

```
#include<ctype.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<signal.h>
#include<stdio.h>
#define SIZE sizeof(struct sockaddr_in)
#define MAX 5
int client[MAX];
int ActiveClients=0;
void findMax(int *maxfd)
{
  int i;
  *maxfd=client[0];
  for(i=1;i<MAX;i++)
    if(client[i] > *maxfd)
      *maxfd=client[i];
}
main()
{
  int sockfd, maxfd,nread,found,i,j;
  char buf[128];
  fd_set fds;
  struct sockaddr_in server = {AF_INET,2000,INADDR_ANY};
  if((sockfd = socket(AF_INET,SOCK_STREAM,0)) == -1)
  {
    printf("Error creating SOCKET\n");
    return(0);
  }
  if( bind(sockfd,(struct sockaddr *)&server,SIZE)==-1)
  {
    printf("bind failed\n");
    return(0);
  }
  if( listen(sockfd,5)==-1)
  {
    printf("listen failed\n");
    return(0);
  }
  findMax(&maxfd);

  for(;;)
  {
```

```
    findMax(&maxfd);
    maxfd=(maxfd>sockfd?maxfd:sockfd)+1;
    FD_ZERO(&fds);
    FD_SET(sockfd,&fds);
    for(i=0;i<MAX;i++)
      if( client[i] != 0 )
        FD_SET(client[i],&fds);
/* Wait for some input or connection request. */
    select(maxfd, &fds,(fd_set *)0,(fd_set *) 0,(struct timeval *) 0);
/*If one of the clients has some input, read and send it to all others.*/
    for(i=0;i<MAX;i++)
      if( FD_ISSET(client[i], &fds))
     {
        nread = recv(client[i], buf, sizeof(buf), 0);
     /* If error or eof, terminate the connection */
        if(nread < 1)
          {
          close(client[i]);
          client[i]=0;
             ActiveClients--;
         }
           else   /* broadcast the message */
             for(j=0;j<MAX;j++)
               if(client[j]!=0 && i!=j)
                send( client[j], buf, nread, 0);
     }
    /* if there is a request for a new connection */
    if( FD_ISSET(sockfd, &fds))
    {
     /* If no of active clients is less than MAX accept the request */
      if(ActiveClients < MAX)
        {
           found = 0;
         for(i=0;i<MAX && !found ;i++ )
          if( client[i]==0 )
          {
               client[i]=accept(sockfd,NULL,NULL);
               found=1;
               ActiveClients++;
           }
     }
    }
  }
}
```