#include<stdio.h> // has the declarations used by standard input and output

#include<sys/types.h> // contains definitions of data types used in system calls by socket.h and in.h

#include<sys/socket.h> // contains definitions of structures needed for sockets

#include<netinet/in.h> // contains constants and structures needed for internet and domain addresses

#include<stdlib.h>

// defines four variable types and several macros,various functions for performing general functions

// socket(int domain,int type,int protocol) this function takes 3 arguments and returns file descriptor

/\* explanation for socket function

domain for ipv4 - AF\_INET

domain for ipv6 - AF\_INET6

type for tcp - SOCK\_STREAM

type for udp - SOCK\_DGRAM

protocal by default 0 for tcp

\*/

/\* int bind(int sockfd,const struct sockadrr \*addr,socklen\_t addrlen)

this function binds filedescriptor generated from above socket function and binds it

return 0 if bind is successfull (or) -1 if failed to bind

sockfd – File descriptor of a socket to be bonded

addr – Structure in which the address to be bound to is specified

addrlen – Size of addr structure

Structure of Socket Address:

struct sockaddr{

sa\_family\_t sa\_family;

char sa\_data[14];

}

\*/

/\*

int listen(int sockfd,int backlog);

backlog - how many connections the system can handle

\*/

/\*

newsockfd=accept(sockfd,(struct sockaddr\*)&addr,&addrlen);

wait for the connection function to call and then execute

\*/

/\*

int connect(int sockfd,struct sockaddr \*addr,socklen\_t addrlen);

return 0 if success

return -1 if failed

\*/

// int read(newsockfd,buffer,buffer\_size);

// int write(newsockfd,buffer,buffer\_size);

/\*

buffer - string we are going to read/write

buffer\_size - size of buffer can't be greater than this

\*/