NAME OF THE PROGRAM: ECE	**DEGREE: B. Tech**
COURSE NAME: Computer Network Lab	**SEMESTER: 6th**
COURSE CODE: EC692	**COURSE CREDIT: 1**
COURSE TYPE: LAB	**CONTACT HOURS:** 0L:0T:2P
NAME OF THE FACULTY: AC, AKS, AT,TP, PCG	

List of experiments:

EC 692 computer network Lab	1. a) Take a decimal number from user. Convert it to different bases (e.g.: 2,8,16 etc.) and send those values to message queue. Write three separate programs to read and display the binary, octal and hex value from the message queue distinctively.	linux, ubuntu
	1. b) Create a message queue. One process will take name and roll number of 'N' students and send those to message queue. Second process will read the names, sort those and send back to message queue; third process will do the same on roll numbers. Then first process will read the entire data and print.	linux, ubuntu
	2. a) Write C Programs to implement a simple client-server application using Unix File socket. The client will send an IPV4 address (X.X.X.X) to the server. The server will verify whether the address is valid or not and send back "Y" or "N" as a result to the client.	linux, ubuntu
	2. b) Write C Programs to implement a simple client-server application. A client will send N integers to the server, which will sort the integers and send back to client. The client will print the result. Use Unix File socket for communication.	linux, ubuntu
	2. c) Write C Programs to implement a simple client-server application. A client will send a string to the server, which will check for palindrome and send back to client. The client will print the result. Use Unix File socket for communication.	linux, ubuntu
	3. a) CAT-5/CAT-6 cable preparation with RJ-45 connector; both straight and cross cabling.	linux, ubuntu
	3. b) IP address configuration (both Static and DHCP) on Linux and Windows systems.	linux, ubuntu
	3. c) Introduction to important network related tools and commands, e.g. ifconfig, ip, hostname, ping, netstat, route, tcpdump, Wireshark, etc.	linux, ubuntu
	4. a)Write C programs to implement TCP Socket. The client will take a bit-stream from the user and send it to the server. The server will implement bit stuffing and send the stream back to the client. The client will print it.	linux, ubuntu
	4. b)Write C programs to implement TCP Socket. The client will take a bit-stream from the user , after taking the bit stream client will stuff it & print it. Then send it to the server. The server will unstuffed it and print the bit stream again.	linux, ubuntu
	5. Create a multi client TCP server.The client will take a bit-stream from the user and send it to server. The server will add a parity bit to it and send the modified bit-stream to the corresponding client.	linux, ubuntu
	6. Create a multi client TCP server.The client will take a dataword and divisor from the user and send them to the server. The server will find out the codeword using CRC and send it back to the corresponding client. The client program will end with special string value "0" as dataword given by the user.	linux, ubuntu

	7. Take a IPV4 address as input. Write a C program to check in which class does it belong. Also print special comment for network ID and broadcast ID.	linux, ubuntu
	8. Write client-server programs using UDP socket. The client will take a data word from the user and send it to the server. The server will find the codeword (use Hamming code error correction technique) and send it back to the client.	linux, ubuntu
	9. Write C programs to implement a simple chat server (single client, single server) by using UDP Socket.	
	10. Write C programs to implement group chat using multicast UDP socket.	