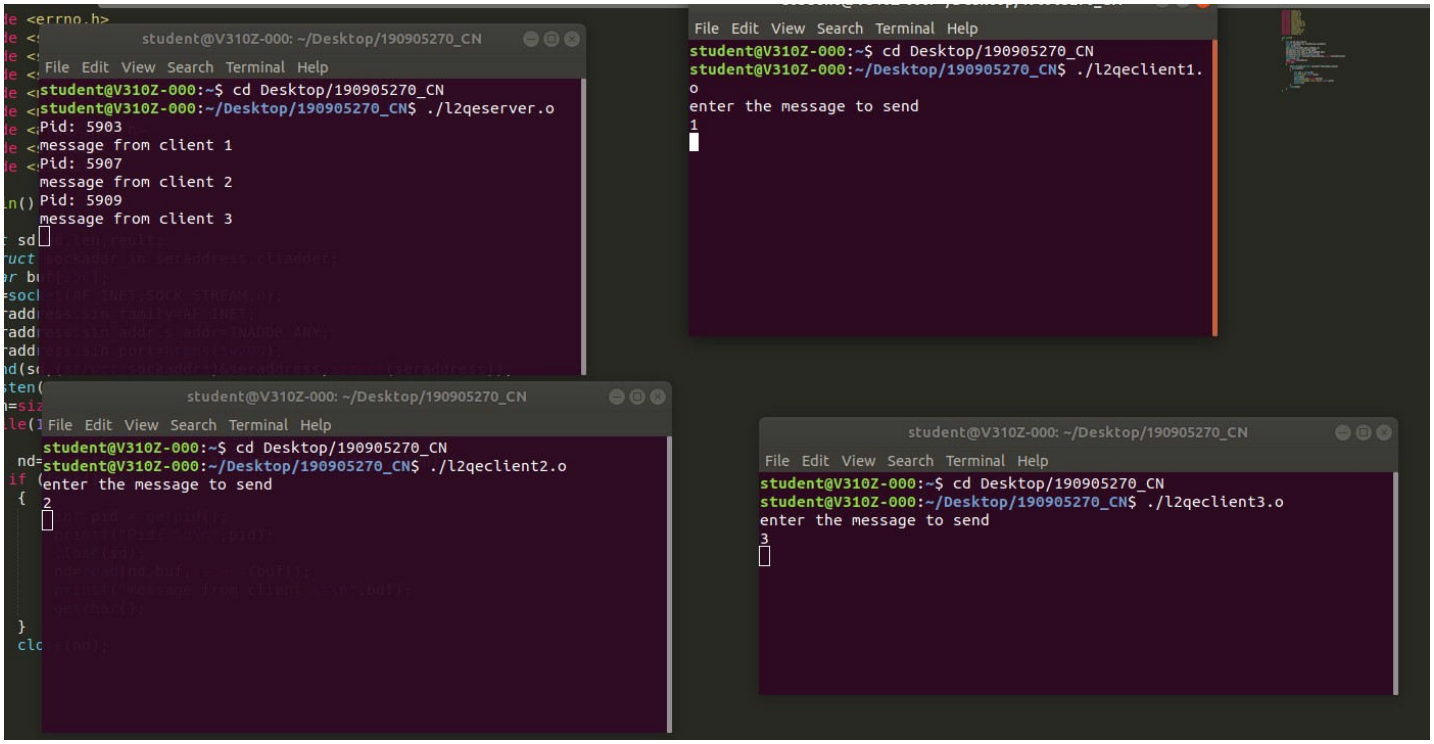
CN LAB 2

Name: B Sreenivas Karthik

Example:

Output:



1)

Code:

Server:

```
#include<stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>
```

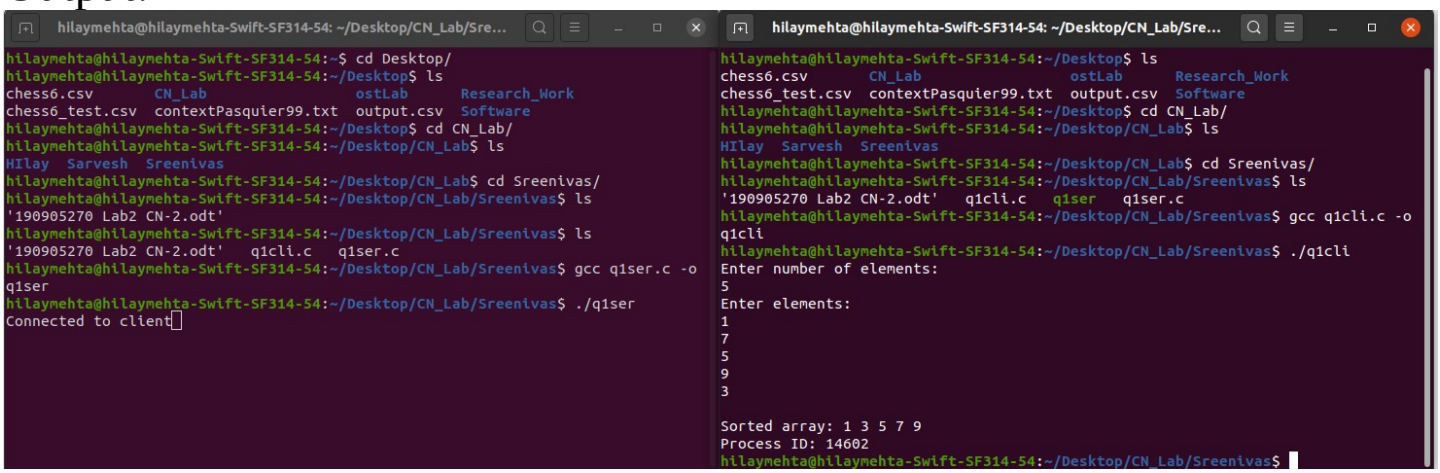
```
int cmpfunc(const void *a, const void *b){
return (*(int *)a - *(int *)b);
}
int main(){
int sd, nd, len, n;
struct sockaddr_in seraddress, cliaddr;
int arr[20];
int arr_size = 0;
sd = socket(AF_INET, SOCK_STREAM, 0);
seraddress.sin_family = AF_INET;
seraddress.sin_addr.s_addr = INADDR_ANY;
seraddress.sin_port = htons(10200);
bind(sd, (struct sockaddr *)&seraddress, sizeof(seraddress));listen(sd, 5);
len = sizeof(cliaddr);
while (1){
nd = accept(sd, (struct sockaddr *)&cliaddr, &len);
printf("Connected to client");
if (fork() == 0){
close(sd);
int pid = getpid();
n = read(nd, &arr_size, sizeof(int));
n = read(nd, arr, arr_size * sizeof(int));
//Sort
qsort(arr, arr_size, sizeof(int), cmpfunc);
n = write(nd, &pid, sizeof(int));
n = write(nd, arr, arr_size * sizeof(int));
getchar();
close(nd);
}
}
}

//Client:

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

```
#include <signal.h>
int main(){
int sd, len, n;
struct sockaddr_in seraddress, cliaddr;
int arr[20];
int arr_size, pid;
sd = socket(AF_INET, SOCK_STREAM, 0);
seraddress.sin_family = AF_INET;
seraddress.sin_addr.s_addr = INADDR_ANY;
seraddress.sin_port = htons(10200);
len = sizeof(seraddress);
connect(sd, (struct sockaddr *)&seraddress, len);
printf("Enter number of elements: \n");
scanf("%d", &arr_size);
printf("Enter elements: \n");
for (int i = 0; i < arr_size; i++){
scanf("%d", &arr[i]);
}
n = write(sd, &arr_size, sizeof(int));
n = write(sd, arr, arr_size * sizeof(int));
n = read(sd, &pid, sizeof(int));
n = read(sd, arr, arr_size * sizeof(int));
printf("\nSorted array: ");
for (int i = 0; i < arr_size; i++){
printf("%d ", arr[i]);
}
printf("\nProcess ID: %d\n", pid);
getchar();
      }
```

Output:



2)
Code:

Server:

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <unistd.h>
#define PORT 5000
int calc(int a, int b, char operator)
{
   switch(operator)
   {
      case '+':
         return a + b;
         break;
      case '-':
         return a - b;
         break;
      case '/':
         return a / b;
         break;
      case '*':
         return a * b;
         break;
      default:
         return 0;
         break;
   }
}
void servfunc(int sockfd, struct sockaddr_in server_address)
{
```

```
    struct sockaddr_in client_address;
    int clientfd, a, b, res, size = sizeof(client_address);
    char op;
    while (1)
    {
        clientfd = accept(sockfd, (struct sockaddr *)&client_address,
 &size);
        if (fork() == 0)
        {
            //in child process
           printf("Child process created with clientfd %d\n",
clientfd);
            close(sockfd);
            read(clientfd, (int *)&a, sizeof(int));
            read(clientfd, (int *)&b, sizeof(int));
            read(clientfd, (char *)&op, sizeof(char));
            res = calc(a, b, op);
            write(clientfd, (int *)&res, sizeof(int));
            close(clientfd);
           printf("Child process terminated with clientfd %d\n",
clientfd);
            exit(0);
        }
        else
            close(clientfd); // parent process
    }
    printf("server closing\n");
 }
 int main()
 {
    int sockfd;
    struct sockaddr_in server_address;
    bzero(&server_address, sizeof(server_address));
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(PORT);
```

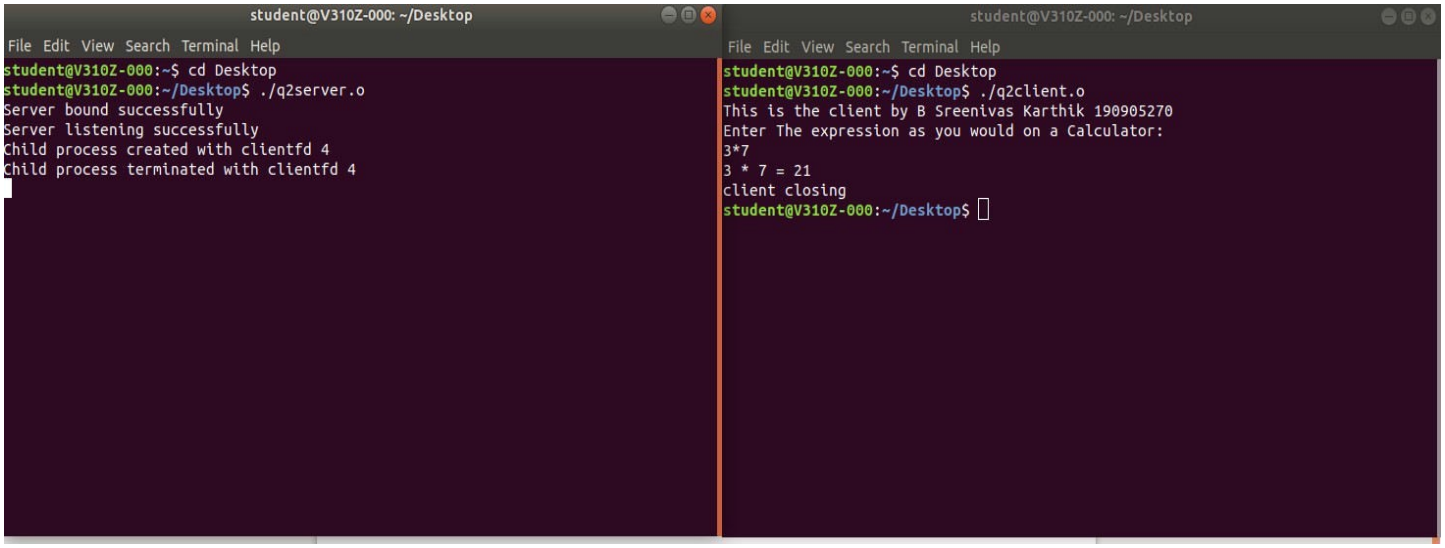
```
    server_address.sin_addr.s_addr = htonl(INADDR_ANY);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
   int res = bind(sockfd, (struct sockaddr *)&server_address,
sizeof(server_address));
    if(res < 0)
    {
       printf("Server unable to bind\n");
       exit(0);
    }
    else
       printf("Server bound successfully\n");
    res = listen(sockfd, 2);
    if(res < 0)
    {
       printf("Server unable to listne\n");
       exit(0);
    }
    else
       printf("Server listening successfully\n");
    servfunc(sockfd, server_address);
    close(sockfd);
 }

 Client:
 #include <sys/types.h>
 #include <sys/socket.h>
 #include <stdio.h>
 #include <netinet/in.h>
 #include <arpa/inet.h>
 #include <unistd.h>
 #include <stdlib.h>
 #define PORT 5000

 void clifunc(int sockfd)
 {
```

```
    printf("This is the client by B Sreenivas Karthik 190905270\n");
    int a, b;
    char c;
    printf("Enter The expression as you would on a Calculator: \n");
    scanf("%d%c%d", &a, &c, &b);
    write(sockfd, (int *)&a, sizeof(int));
    write(sockfd, (int *)&b, sizeof(int));
    write(sockfd, (char *)&c, sizeof(char));
    int res;
    read(sockfd, (int *)&res, sizeof(int));
    printf("%d %c %d = %d\n", a, c, b, res);
    printf("client closing\n");
 }
 int main(int argc, char const *argv[])
 {
    int sockfd;
    int len;
    struct sockaddr_in server_address;
    int result;
    char ch;
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    bzero(&server_address, sizeof(server_address));
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(PORT);
    server_address.sin_addr.s_addr = htonl(INADDR_ANY);
    len = sizeof(server_address);
   result = connect(sockfd, (struct sockaddr *)&server_address,
len);
    if(result == -1)
    {
       printf("connection error\n");
       exit(0);
    }
    clifunc(sockfd);
    close(sockfd);
```

}


student@V310Z-000: ~/Desktop
File Edit View Search Terminal Help
student@V310Z-000:~$ cd Desktop
student@V310Z-000:~/Desktop$ ./q2server.o
Server bound successfully
Server listening successfully
Child process created with clientfd 4
Child process terminated with clientfd 4
student@V310Z-000: ~/Desktop
File Edit View Search Terminal Help
student@V310Z-000:~$ cd Desktop
student@V310Z-000:~/Desktop$ ./q2client.o
This is the client by B Sreenivas Karthik 190905270
Enter The expression as you would on a Calculator:
3*7
3 * 7 = 21
client closing
student@V310Z-000:~/Desktop$