Shatabdi Chakraborty
Regd no. 1841017152
Experiment no.2

Objective-1

Program using TCP sockets to transfer numerical data between client and server.Then server squares the number and sends to client.

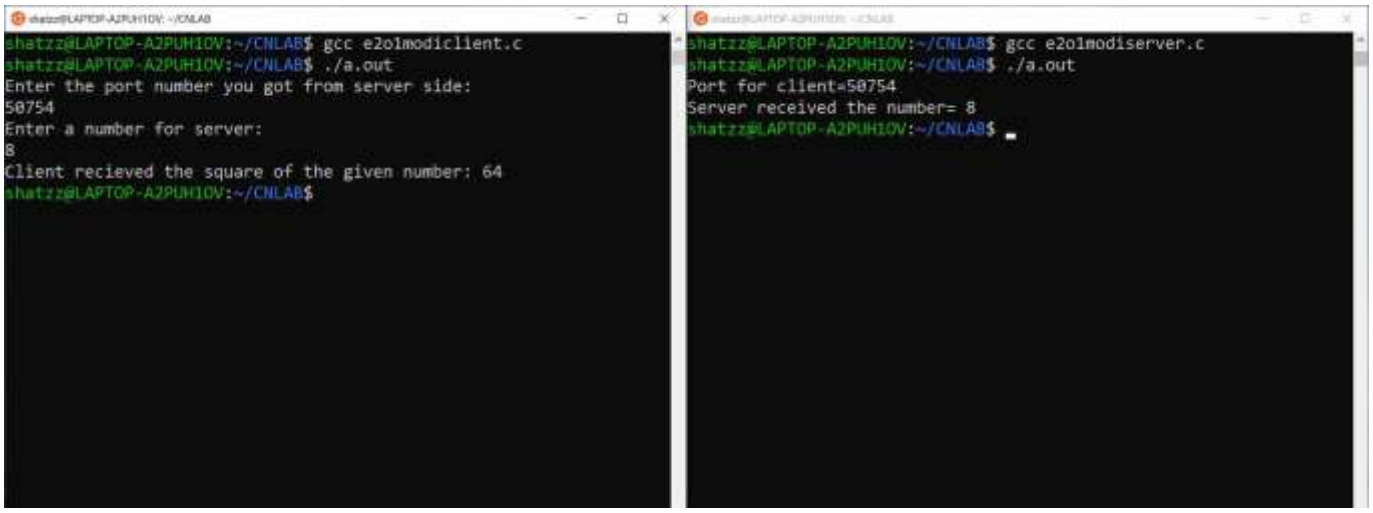
Server-

```
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
int main()
{
        int sockfd, len, n;
         long port;
         char buff[200];
         struct sockaddr_in cliaddr;
         len = sizeof(cliaddr);
         printf("Enter the port number you got from server side:
\n");
         scanf("%ld", &port);
         cliaddr.sin_family = AF_INET;
         cliaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
         cliaddr.sin_port = htons(port);
         sockfd = socket(AF_INET, SOCK_STREAM, 0);
         connect(sockfd, (struct sockaddr *)&cliaddr, len);
         printf("Enter a number for server: \n");
         scanf("%s", buff);
         write(sockfd, buff, sizeof(buff));
         n = read(sockfd, buff, sizeof(buff));
         buff[n] = 0;
         int num = atoi(buff);
         printf("Client recieved the square of the given number:
%d\n", num);
 return 0;
}
```

Client-

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

int main()
{
        int listenfd, connfd, len, n;
         char buff[200];
         struct sockaddr_in servaddr, cliaddr;
         len = sizeof(servaddr);
         servaddr.sin_family = AF_INET;

         servaddr.sin_addr.s_addr = htons(INADDR_ANY);

         servaddr.sin_port = htons(0);
         listenfd = socket(AF_INET, SOCK_STREAM, 0);
         bind(listenfd, (struct sockaddr *)&servaddr, len);
         getsockname(listenfd, (struct sockaddr *)&servaddr, &len);
         printf("Port for client=%ld\n", (long)ntohs(servaddr.sin_port));
         listen(listenfd, 5);
         connfd = accept(listenfd, (struct sockaddr *)&cliaddr, &len);
         n = read(connfd, buff, sizeof(buff));

         buff[n] = 0;
         int num = atoi(buff);

         printf("Server received the number= %d\n", num);
         int numdata[1];
         numdata[0] = num * num;
          strcpy(buff, "");
         for (int i = 0; i < 2; i++)
          {
                 char temp[50];
                 sprintf(temp, "%d", numdata[i]);
                 strcat(buff, temp);
                  strcat(buff, " ");
         }
         write(connfd, buff, sizeof(buff));

 return 0;
}
```

Output-



Objective-2

To implement a program where the client reads 10 numbers and sends to the server. The server sort them and displays them at the client end.

Server-

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
int compare(const void * a, const void * b)
{
        return ( *(int*)a - *(int*)b );
}
int main()
{
        int listenfd,connfd,len,n;
        char buff[200], data[500];
        struct sockaddr_in servaddr,cliaddr;
        len=sizeof(servaddr);
        servaddr.sin_family=AF_INET;
        servaddr.sin_addr.s_addr=htons( INADDR_ANY);
        servaddr.sin_port=htons(0);
        listenfd=socket(AF_INET,SOCK_STREAM,0);
        bind(listenfd, (struct sockaddr *)&servaddr,len);
        getsockname(listenfd,(struct sockaddr *)&servaddr ,&len);
        printf("Port for client=%ld\n",(long)ntohs(servaddr.sin_port));
        listen(listenfd,5);
        connfd=accept(listenfd,(struct sockaddr *)&cliaddr ,&len);
        n=read(connfd , buff ,sizeof(buff));
        buff[n]=0;
        int num = atoi(buff);
```

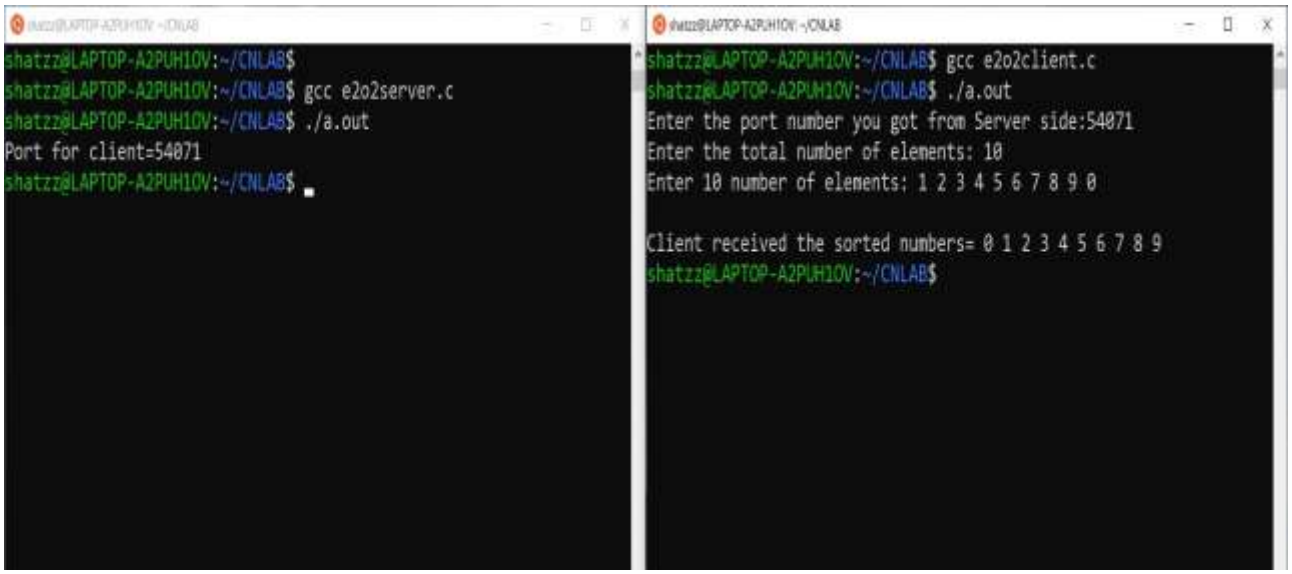
```
        n=read(connfd , data ,sizeof(data));
        data[n]=0;
        int arr[num], i = 0;
        char* temp = strtok(data, " ");
        while (temp != NULL)
        {
                arr[i++] = atoi(temp);
                temp = strtok(NULL, " ");
        }
        qsort(arr, num, sizeof(int), compare);
        strcpy(data, "");
        for(int i = 0; i < num; i++)
        {
                char temp[50];
                sprintf (temp, "%d", arr[i]);
                strcat(data, temp);
                strcat(data, " ");
        }
        write(connfd, data ,sizeof(data));
        return 0;
}
```

Client-

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
int main()
{
        int sockfd,len,n;
        long port;
        char buff[200], data[500] = { '\0' };
        struct sockaddr_in cliaddr;
        len=sizeof(cliaddr);
        printf("Enter the port number you got from Server side:");
        scanf("%ld" ,&port);
        cliaddr.sin_family=AF_INET;
        cliaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
        cliaddr.sin_port=htons(port);
        sockfd=socket(AF_INET,SOCK_STREAM,0);
        connect(sockfd,(struct sockaddr *)&cliaddr,len);
        printf("Enter the total number of elements: ");
        scanf("%s", buff);
        int num = atoi(buff);
        write(sockfd, buff ,sizeof(buff));
```

```
        printf("Enter %d number of elements: ", num);
        for(int i = 0; i < num; i++)
        {
                scanf("%s", buff);
                strcat(data, buff);
                strcat(data, " ");
        }
        write(sockfd, data ,sizeof(data));
        n=read(sockfd,data ,sizeof(data));
        data[n]=0;
        printf("\nClient received the sorted numbers= %s\n" , data
);
        return 0;
}
```

Output-



Objective-3

To implement a program where a clients reads a number x and sends to the server , the server sends to 2x and x^2 to the cllient.The client adds them and sends the result to the server , the server displays it.

Server-

```
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
```

```c
#include <netinet/in.h>
int main()
{
        int listenfd,connfd,len,n;
        char buff[200];
        struct sockaddr_in servaddr,cliaddr;
        len=sizeof(servaddr);
        servaddr.sin_family=AF_INET;
        servaddr.sin_port=htons(0);
        listenfd=socket(AF_INET,SOCK_STREAM,0);
        bind(listenfd, (struct sockaddr *)&servaddr,len);
        getsockname(listenfd,(struct sockaddr *)&servaddr ,&len);
        printf("Port for client=%ld\n",(long)ntohs(servaddr.sin_port));
        listen(listenfd,5);
        connfd=accept(listenfd,(struct sockaddr *)&cliaddr ,&len);
        // Read the number from client
         n=read(connfd , buff ,sizeof(buff));
         buff[n]=0;
         int num = atoi(buff);
         printf("\nReceived number from client= %d\n", num);

         int numdata[2];
         numdata[0] = num * 2;
         numdata[1] = num * num;
         strcpy(buff, "");
         for(int i = 0; i < 2; i++)
         {
         char temp[50];
         sprintf (temp, "%d", numdata[i]);
         strcat(buff, temp);
         strcat(buff, " ");
         }

         write(connfd, buff ,sizeof(buff));

         n=read(connfd , buff ,sizeof(buff));
         buff[n]=0;
         printf("\nReceived sum from client: %s\n", buff);
         return 0;
         }
```

Client-

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

```c
#include <netinet/in.h>
int main()
{
        int sockfd,len,n;
        long port;
        char buff[200];
        struct sockaddr_in cliaddr;
        len=sizeof(cliaddr);
        printf("Enter the port number you got from Server side:");
        scanf("%ld" ,&port);
        cliaddr.sin_family=AF_INET;
        cliaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
        cliaddr.sin_port=htons(port);
        sockfd=socket(AF_INET,SOCK_STREAM,0);
        connect(sockfd,(struct sockaddr *)&cliaddr,len);

         printf("\nEnter a number: ");
         scanf("%s", buff);
         write(sockfd, buff ,sizeof(buff));

         n=read(sockfd,buff ,sizeof(buff));
         buff[n]=0;
         printf("\nReceived 2n and n^2 from server: %s\n", buff);

         char* temp;
         int sum = 0;
         temp = strtok(buff, " ");
         while( temp != NULL )
         {
          sum += atoi(temp);
           temp = strtok(NULL, " ");
            }

              strcpy(buff, "");
              sprintf (buff, "%d", sum);
              write(sockfd, buff ,sizeof(buff));
              return 0;
  }
```

Output-