**Name – Akash Kumar Singh**

**Reg. No – 1841017025**

**Branch – CSIT**

**Sec – A**

LAB -2

Q1>

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

Server

#include <stdio.h>

#include <unistd.h>

#include <string.h>

#include <stdlib.h>

#include <errno.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#define MAXBUF 1024

#define QUEUE\_NO 5

int main(int argc, char \*\*argv) {

    int sockfd, clientfd, port, int\_to\_receive,int\_to\_send;

    struct sockaddr\_in s\_addr, c\_addr;

    char buffer[MAXBUF];

    socklen\_t socklen = (socklen\_t)sizeof(struct sockaddr\_in);

    if (argc < 2) {

        printf("Some of the command line arguments missing");

        return -1;

    }

    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (sockfd == -1) {

        printf("socket creation failed...\n");

        exit(errno);

    }

        printf("Socket successfully created..\n");

    bzero(&s\_addr, sizeof(s\_addr));

    s\_addr.sin\_family = AF\_INET;

    port = atoi(argv[1]);

    s\_addr.sin\_port = htons(port);

    s\_addr.sin\_addr.s\_addr = INADDR\_ANY;

    if (bind(sockfd, (struct sockaddr\*)&s\_addr, sizeof(s\_addr)) != 0){

        printf("socket creation failed...\n");

        exit(errno);

    }

    if (listen(sockfd, QUEUE\_NO) != 0) {

        printf("socket listen failed...\n");

        exit(errno);

    }

    while (1) {

        clientfd = accept(sockfd, (struct sockaddr\*)&c\_addr, &socklen);

        printf("%s:%d connected\n", inet\_ntoa(c\_addr.sin\_addr), ntohs(c\_addr.sin\_port));

        read(clientfd, &int\_to\_receive, sizeof(int\_to\_receive));

*//int\_to\_receive = ntohl(int\_to\_receive);*

        int\_to\_send=int\_to\_receive\*int\_to\_receive;

        write(clientfd, &int\_to\_send, sizeof(int\_to\_send));

        printf("Received from client: %d\n", int\_to\_receive);

        close(clientfd);

    }

    close(sockfd);

    return 0;

}

Client

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <errno.h>

void send\_receive\_integer\_over\_socket(int sockfd) {

    int int\_to\_send;

    printf("enter the integer value to be sent to server\n");

    scanf("%d",&int\_to\_send);

    int int\_to\_receive;

    write(sockfd, &int\_to\_send, sizeof(int\_to\_send));

    read(sockfd, &int\_to\_receive, sizeof(int\_to\_receive));

    printf("Integer Received from Server : %d\n", int\_to\_receive);

}

int main(int argc, char \*\*argv) {

    int sockfd, port;

    struct sockaddr\_in servaddr, cli;

    if (argc < 3) {

        printf("Some of the command line arguments missing");

        return -1;

    }

    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (sockfd == -1) {

        printf("socket creation failed...\n");

        exit(errno);

    }

    printf("Socket successfully created..\n");

    bzero(&servaddr, sizeof(servaddr));

    servaddr.sin\_family = AF\_INET;

    servaddr.sin\_addr.s\_addr = inet\_addr(argv[1]);

    port = atoi(argv[2]);

    servaddr.sin\_port = htons(port);

    if (connect(sockfd, (struct sockaddr\*)&servaddr, sizeof(servaddr)) != 0) {

        printf("connection with the server failed...\n");

        exit(errno);

    }

    printf("connected to the server..\n");

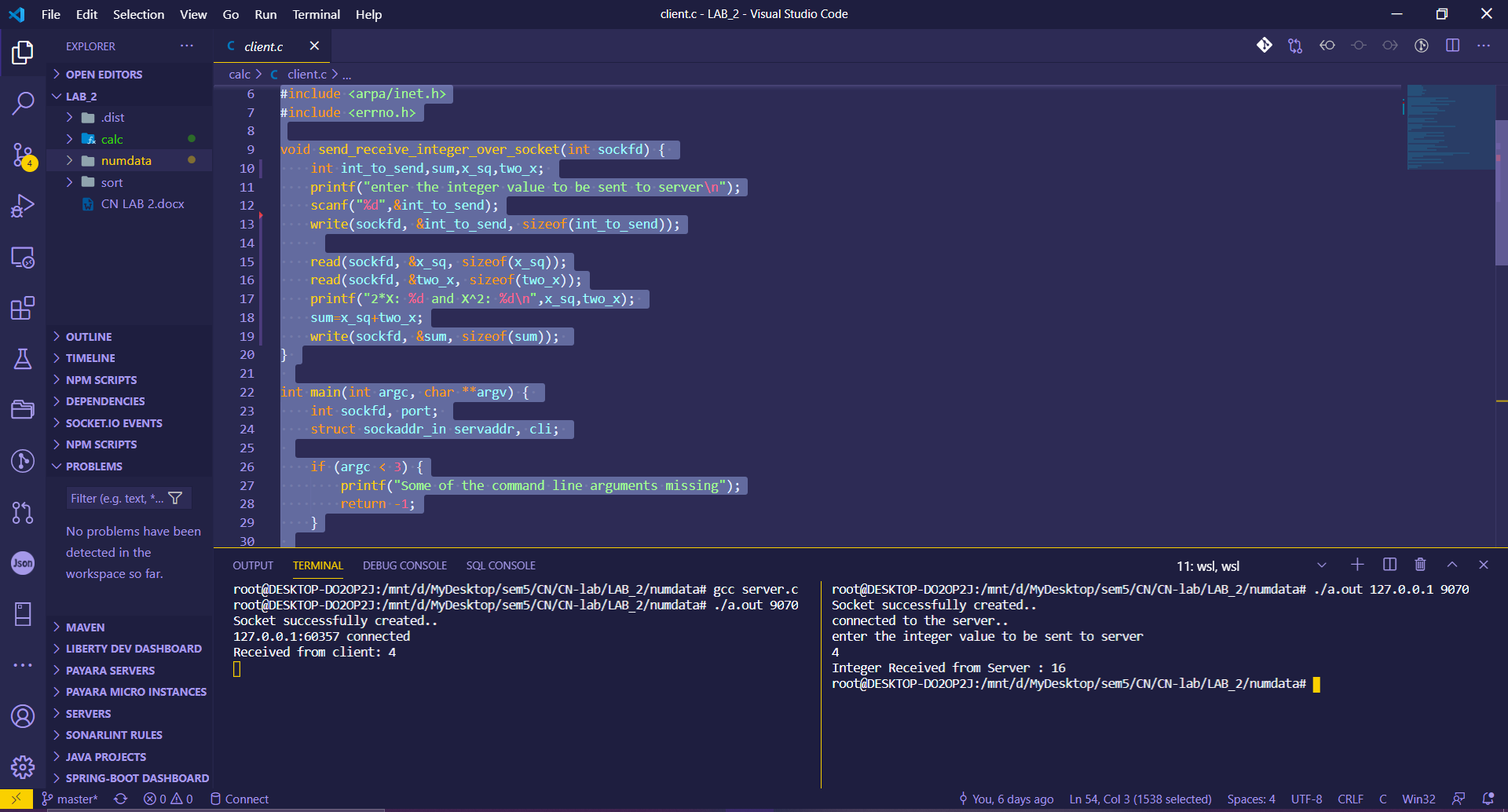
    send\_receive\_integer\_over\_socket(sockfd);

    close(sockfd);

    return 0;

}

Output



Q2>

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

Server

*// Server code in C to sort the array*

#include <arpa/inet.h>

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <unistd.h>

void bubble\_sort(int*[]*, int);

*// Driver code*

int main(int argc, char\* argv*[]*)

{

    int socket\_desc, client\_sock, c, read\_size;

    struct sockaddr\_in server, client;

    int message[10], i;

*// Create socket*

    socket\_desc = socket(AF\_INET, SOCK\_STREAM, 0);

    if (socket\_desc == -1) {

        printf("Could not create socket");

    }

    puts("Socket creation done...");

*// Prepare the sockaddr\_in structure*

    server.sin\_family = AF\_INET;

    server.sin\_addr.s\_addr = INADDR\_ANY;

    server.sin\_port = htons(8880);

*// Bind the socket*

    if (bind(socket\_desc, (struct sockaddr\*)&server, sizeof(server)) < 0) {

*// print the error message*

        perror("bind failed. Error");

        return 1;

    }

    puts("Server is binded...");

*// lsiten to the socket*

    listen(socket\_desc, 3);

    puts("Waiting for client to connect\n");

    c = sizeof(struct sockaddr\_in);

*// accept connection from an incoming client*

    client\_sock = accept(socket\_desc, (struct sockaddr\*)&client, (socklen\_t\*)&c);

    if (client\_sock < 0) {

        perror("accept failed");

        return 1;

    }

    puts("Connection accepted");

*// Receive a message from client*

    while ((read\_size = recv(client\_sock, &message, 10 \* sizeof(int), 0)) > 0) {

        bubble\_sort(message, 10);

        write(client\_sock, &message, 10 \* sizeof(int));

    }

    if (read\_size == 0) {

        puts("Client disconnected");

    }

    else if (read\_size == -1) {

        perror("recv failed");

    }

    return 0;

}

*// Function to sort the array*

void bubble\_sort(int list*[]*, int n)

{

    int c, d, t;

    for (c = 0; c < (n - 1); c++) {

        for (d = 0; d < n - c - 1; d++) {

            if (list[d] > list[d + 1]) {

*/\* Swapping \*/*

                t = list[d];

                list[d] = list[d + 1];

                list[d + 1] = t;

            }

        }

    }

}

Client

*// Client code in C to sort the array*

#include <arpa/inet.h>

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <unistd.h>

int main(int argc, char\* argv*[]*)

{

    int sock;

    struct sockaddr\_in server;

    int size=10;

    int server\_reply[size];

    int number[size], i, temp;

    printf("enter the array elements\n");

    for(int x=0;x<size;x++)

    {

        scanf("%d",&number[x]);

    }

    sock = socket(AF\_INET, SOCK\_STREAM, 0);

    if (sock == -1) {

        printf("Could not create socket");

    }

    puts("Socket created");

    server.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

    server.sin\_family = AF\_INET;

    server.sin\_port = htons(8880);

    if (connect(sock, (struct sockaddr\*)&server, sizeof(server)) < 0) {

        perror("connect failed. Error");

        return 1;

    }

    puts("Connection to server was successful\n");

    if (send(sock, &number, size \* sizeof(int), 0) < 0) {

        puts("Send failed");

        return 1;

    }

    if (recv(sock, &server\_reply, size \* sizeof(int), 0) < 0) {

        puts("recv failed");

        return 0;

    }

    puts("Sorted numbers:\n");

    for (i = 0; i < size; i++) {

        printf("%d ", server\_reply[i]);

    }

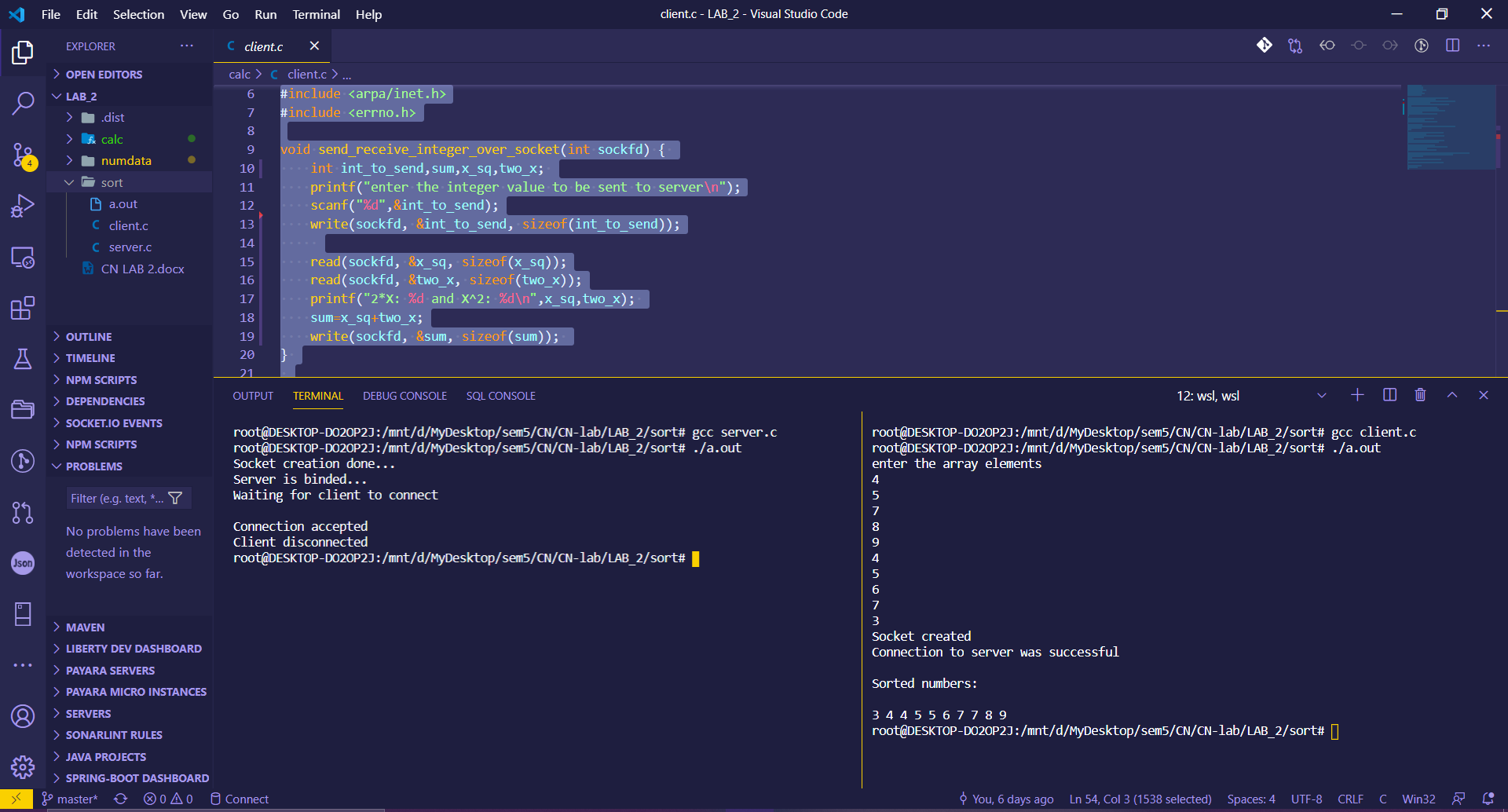
    printf("\n");

    close(sock);

    return 0;

}

Output



Q3>

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 <stdio.h>

#include <unistd.h>

#include <string.h>

#include <stdlib.h>

#include <errno.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#define MAXBUF 1024

#define QUEUE\_NO 5

int main(int argc, char \*\*argv) {

    int sockfd, clientfd, port, int\_to\_receive,int\_to\_send,two\_x,x\_sq,sum;

    struct sockaddr\_in s\_addr, c\_addr;

    char buffer[MAXBUF];

    socklen\_t socklen = (socklen\_t)sizeof(struct sockaddr\_in);

    if (argc < 2) {

        printf("Some of the command line arguments missing");

        return -1;

    }

    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (sockfd == -1) {

        printf("socket creation failed...\n");

        exit(errno);

    }

        printf("Socket successfully created..\n");

    bzero(&s\_addr, sizeof(s\_addr));

    s\_addr.sin\_family = AF\_INET;

    port = atoi(argv[1]);

    s\_addr.sin\_port = htons(port);

    s\_addr.sin\_addr.s\_addr = INADDR\_ANY;

    if (bind(sockfd, (struct sockaddr\*)&s\_addr, sizeof(s\_addr)) != 0){

        printf("socket creation failed...\n");

        exit(errno);

    }

    if (listen(sockfd, QUEUE\_NO) != 0) {

        printf("socket listen failed...\n");

        exit(errno);

    }

    clientfd = accept(sockfd, (struct sockaddr\*)&c\_addr, &socklen);

    printf("%s:%d connected\n", inet\_ntoa(c\_addr.sin\_addr), ntohs(c\_addr.sin\_port));

    read(clientfd, &int\_to\_receive, sizeof(int\_to\_receive));

    printf("Received from client: %d\n", int\_to\_receive);

    x\_sq=int\_to\_receive\*int\_to\_receive;

    two\_x=2\*int\_to\_receive;

    write(clientfd, &two\_x, sizeof(two\_x));

    write(clientfd, &x\_sq, sizeof(x\_sq));

    read(clientfd, &sum, sizeof(sum));

    printf("sum of X^2 and 2\*X recieved from client = %d\n",sum);

    close(clientfd);

    close(sockfd);

    return 0;

}

Client

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <errno.h>

void send\_receive\_integer\_over\_socket(int sockfd) {

    int int\_to\_send,sum,x\_sq,two\_x;

    printf("enter the integer value to be sent to server\n");

    scanf("%d",&int\_to\_send);

    write(sockfd, &int\_to\_send, sizeof(int\_to\_send));

    read(sockfd, &x\_sq, sizeof(x\_sq));

    read(sockfd, &two\_x, sizeof(two\_x));

    printf("2\*X: %d and X^2: %d\n",x\_sq,two\_x);

    sum=x\_sq+two\_x;

    write(sockfd, &sum, sizeof(sum));

}

int main(int argc, char \*\*argv) {

    int sockfd, port;

    struct sockaddr\_in servaddr, cli;

    if (argc < 3) {

        printf("Some of the command line arguments missing");

        return -1;

    }

    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (sockfd == -1) {

        printf("socket creation failed...\n");

        exit(errno);

    }

    printf("Socket successfully created..\n");

    bzero(&servaddr, sizeof(servaddr));

    servaddr.sin\_family = AF\_INET;

    servaddr.sin\_addr.s\_addr = inet\_addr(argv[1]);

    port = atoi(argv[2]);

    servaddr.sin\_port = htons(port);

    if (connect(sockfd, (struct sockaddr\*)&servaddr, sizeof(servaddr)) != 0) {

        printf("connection with the server failed...\n");

        exit(errno);

    }

    printf("connected to the server..\n");

    send\_receive\_integer\_over\_socket(sockfd);

    close(sockfd);

    return 0;

}

Output

