COOCHBEHAR GOVERMENT ENGINEERING COLLEGE

COMPUTER NETWORKS LAB

Nikti Paul

Roll: 34900119032
6th Semester, CSE

1. Write a client server program to communicate between them (two way).

Client code

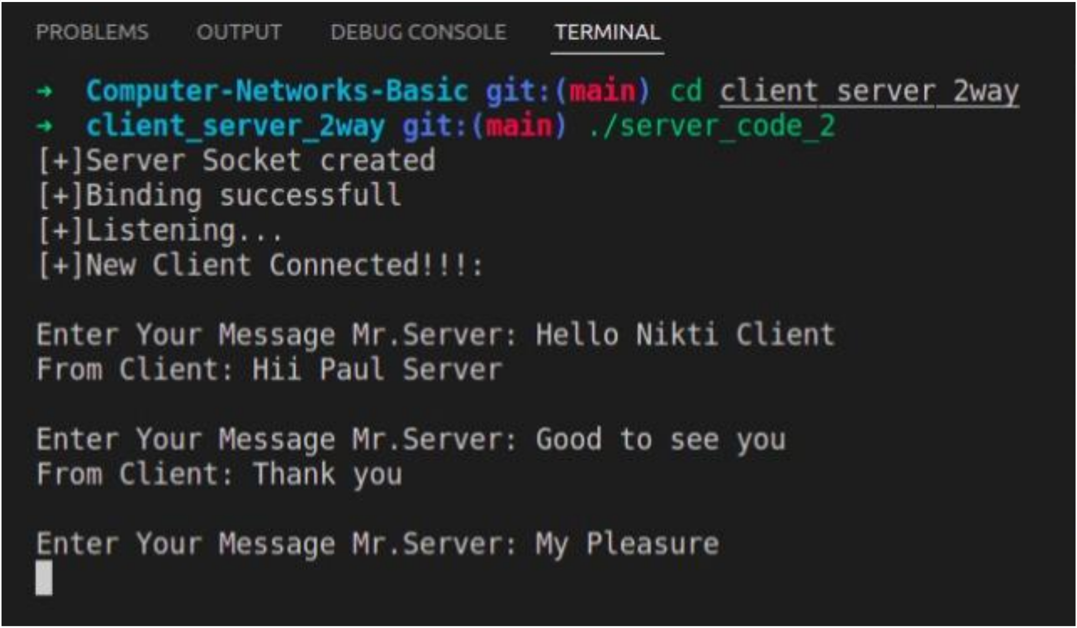
```
#include<stdio.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<string.h>
int main()
{
        struct sockaddr_in c_addr;
        int c_fd,c_len;
        char buff[100];

        if((c_fd=socket(AF_INET,SOCK_STREAM,0))==-1)
            printf("[-]Error in Socket\n");
        printf("[+]Client Socket created\n");

        c_addr.sin_family=AF_INET;
        c_addr.sin_addr.s_addr=INADDR_ANY;
        c_addr.sin_port=3452;
        c_len=sizeof(c_addr);

        if(connect(c_fd,(struct sockaddr*)& c_addr,c_len)==-1)
            printf("[-]Error in Connect\n");
        printf("[+]Connected to the Server: \n\n");

        while(1)
        {
            read(c_fd,buff,100);
            printf("From Server: %s\n",buff);
            printf("Enter your Message Mr.Client: ");
            fgets(buff,sizeof(buff),stdin);

            if (strcmp(buff, "Exit\n") == 0){
                printf("[-]Disconneted from Server");
                write(c_fd,"Disconnected\n",100);
                break;
            }
            else{
                write(c_fd,buff,100);
            }
        }
        close(c_fd);
        return 0;
}
```

Server code

```c
#include<stdio.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>
int main()
{
    struct sockaddr_in s_addr,c_addr;

    int s_fd,c_fd,s_len,c_len;



    if((s_fd=socket(AF_INET,SOCK_STREAM,0))==-1)
        printf("[-]Error in Socket\n");
    printf("[+]Server Socket created\n");

    s_addr.sin_family = AF_INET;
    s_addr.sin_port = 3452;
    s_len=sizeof(s_addr);

    if(bind(s_fd,(struct sockaddr*)& s_addr,s_len)==-1)
        printf("[-]Error in binding\n");
    printf("[+]Binding successfull\n");

    if(listen(s_fd,5)==-1)
        printf("[-]Error in listen\n");
    printf("[+]Listening... \n");


    c_len=sizeof(c_addr);
    if((c_fd=accept(s_fd,(struct sockaddr*)&c_addr,&c_len))==-1)
        printf("\n[-]Error in accepting\n");
    printf("[+]New Client Connected!!!: \n\n");

    char buff[100];
    while(1)
    {
        printf("Enter Your Message Mr.Server: ");
        fgets(buff,sizeof(buff),stdin);
        write(c_fd,buff,100);
        read(c_fd,buff,100);
        printf("From Client: %s\n",buff);

    }
    close(c_fd);
    return 0;
}
```

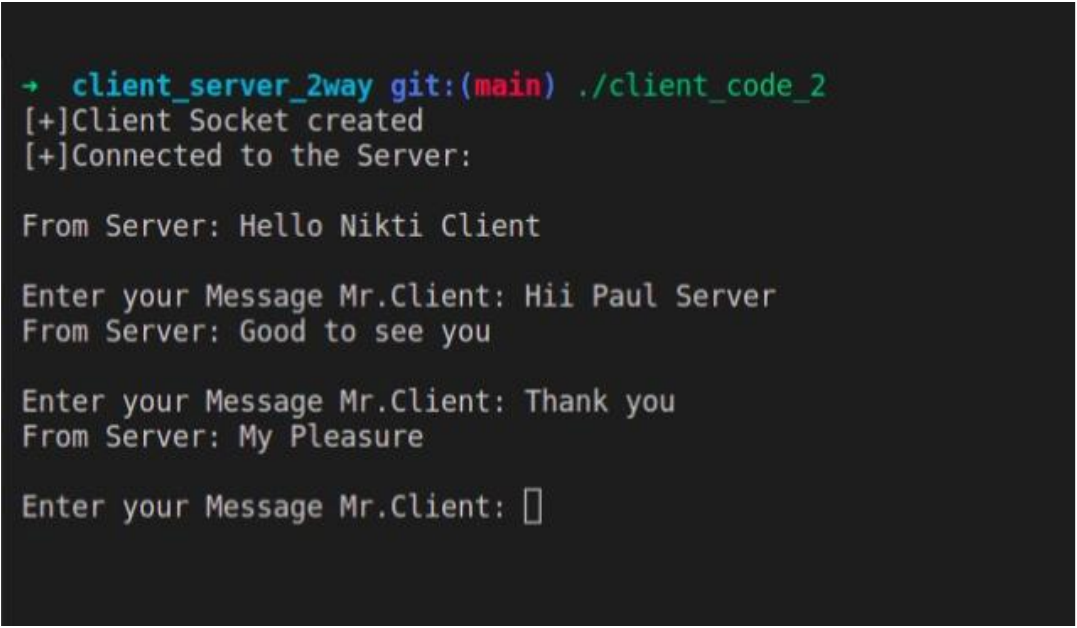
Output:

Server:



```
→  Computer-Networks-Basic git:(main) cd client_server_2way
→  client_server_2way git:(main) ./server_code_2
[+]Server Socket created
[+]Binding successfull
[+]Listening...
[+]New Client Connected!!!:

Enter Your Message Mr.Server: Hello Nikti Client
From Client: Hii Paul Server

Enter Your Message Mr.Server: Good to see you
From Client: Thank you

Enter Your Message Mr.Server: My Pleasure
```

Client:



```
→  client_server_2way git:(main) ./client_code_2
[+]Client Socket created
[+]Connected to the Server:

From Server: Hello Nikti Client

Enter your Message Mr.Client: Hii Paul Server
From Server: Good to see you

Enter your Message Mr.Client: Thank you
From Server: My Pleasure

Enter your Message Mr.Client:
```

Code Links (from my GITHUB account):

- Client Code
- Server Code

2. Write a program that will ask the client program to enter two numbers and the server will display the addition, subtraction, and multiplication of that two numbers.

Client code

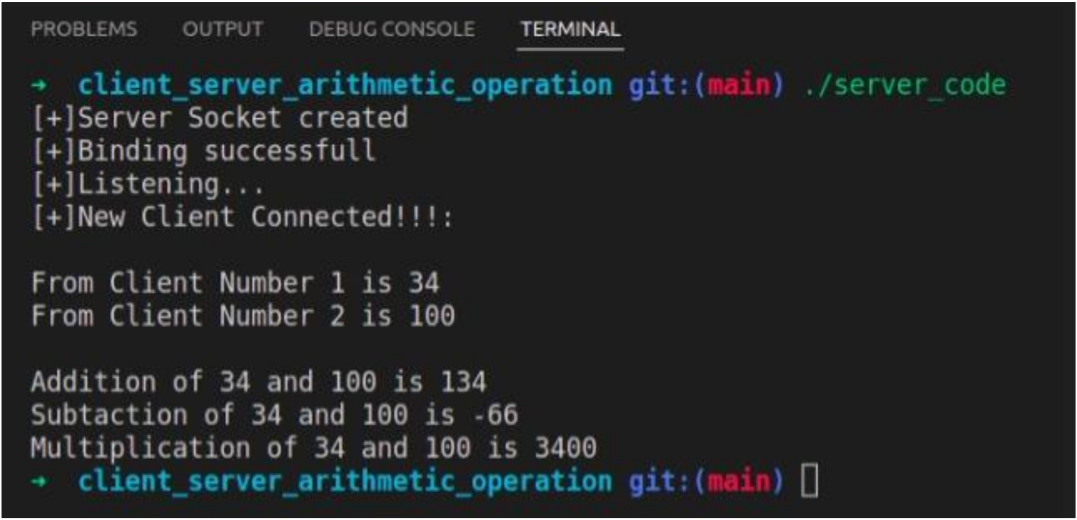
```
#include<stdio.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<string.h>
#include<stdlib.h>

int main()
{
        struct sockaddr_in c_addr;
        int c_fd,c_len;
        char buff[100];

        if((c_fd=socket(AF_INET,SOCK_STREAM,0))==-1)
            printf("[-]Error in Socket\n");
        printf("[+]Client Socket created\n");

        c_addr.sin_family=AF_INET;
        c_addr.sin_addr.s_addr=INADDR_ANY;
        c_addr.sin_port=3452;
        c_len=sizeof(c_addr);

        if(connect(c_fd,(struct sockaddr*)& c_addr,c_len)==-1)
            printf("[-]Error in Connect\n");
        printf("[+]Connected to the Server: \n\n");

        int n = 0;
        while(n < 2){
            printf("Enter Number %d: ",n+1);
            fgets(buff,sizeof(buff),stdin);
            write(c_fd,buff,100);
            n++;
        }
        close(c_fd);
        return 0;
}
```

Server code

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

int main()
{
    struct sockaddr_in s_addr,c_addr;

    int s_fd,c_fd,s_len,c_len;



    if((s_fd=socket(AF_INET,SOCK_STREAM,0))==-1)
        printf("[-]Error in Socket\n");
    printf("[+]Server Socket created\n");


    s_addr.sin_family = AF_INET;
    s_addr.sin_port = 3452;
    s_len=sizeof(s_addr);

    if(bind(s_fd,(struct sockaddr*)& s_addr,s_len)==-1)
        printf("[-]Error in binding\n");
    printf("[+]Binding successfull\n");

    if(listen(s_fd,5)==-1)
        printf("[-]Error in listen\n");
    printf("[+]Listening... \n");



    int nums[2], n = 0;

    c_len=sizeof(c_addr);
    if((c_fd=accept(s_fd,(struct sockaddr*)&c_addr,&c_len))==-1)
        printf("\n[-]Error in accepting\n");
    printf("[+]New Client Connected!!!: \n\n");


    while(n < 2)
    {

        char buff[100];
        read(c_fd,buff,100);

        printf("From Client Number %d is %s",n+1,buff);
        nums[n++] = atoi(buff);

    }

    printf("\nAddition of %d and %d is %d\n",nums[0],nums[1],nums[0]+nums[1]);
    printf("Subtaction of %d and %d is %d\n",nums[0],nums[1],nums[0]-nums[1]);
    printf("Multiplication of %d and %d is %d\n",nums[0],nums[1],nums[0]*nums[1]);

    close(c_fd);
    return 0;
}
```

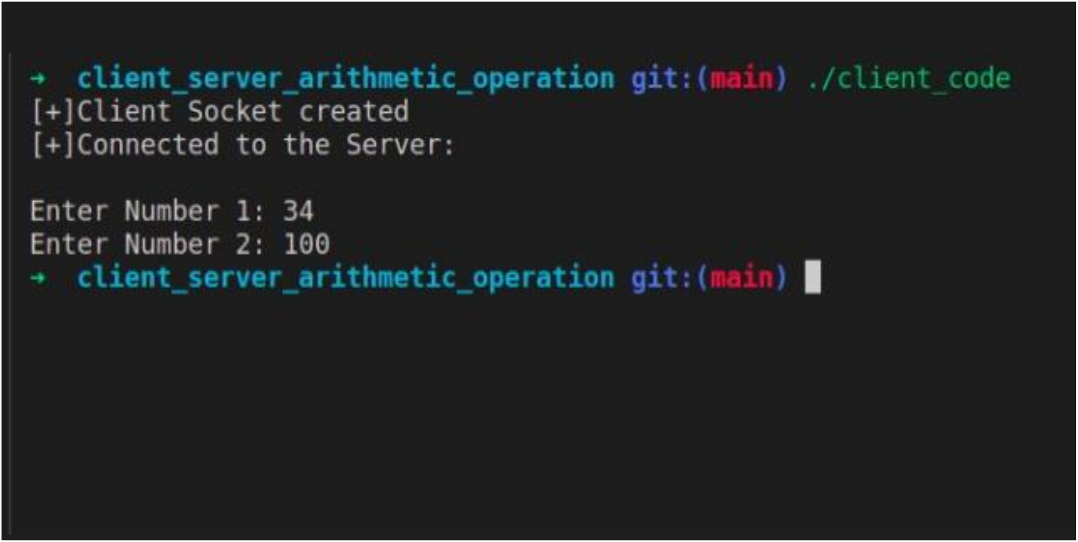
Output:

Server:



```
→  client_server_arithmetic_operation git:(main) ./server_code
[+]Server Socket created
[+]Binding successfull
[+]Listening...
[+]New Client Connected!!!:

From Client Number 1 is 34
From Client Number 2 is 100

Addition of 34 and 100 is 134
Subtaction of 34 and 100 is -66
Multiplication of 34 and 100 is 3400
→  client_server_arithmetic_operation git:(main) 
```

Client:



```
→  client_server_arithmetic_operation git:(main) ./client_code
[+]Client Socket created
[+]Connected to the Server:

Enter Number 1: 34
Enter Number 2: 100
→  client_server_arithmetic_operation git:(main) 
```

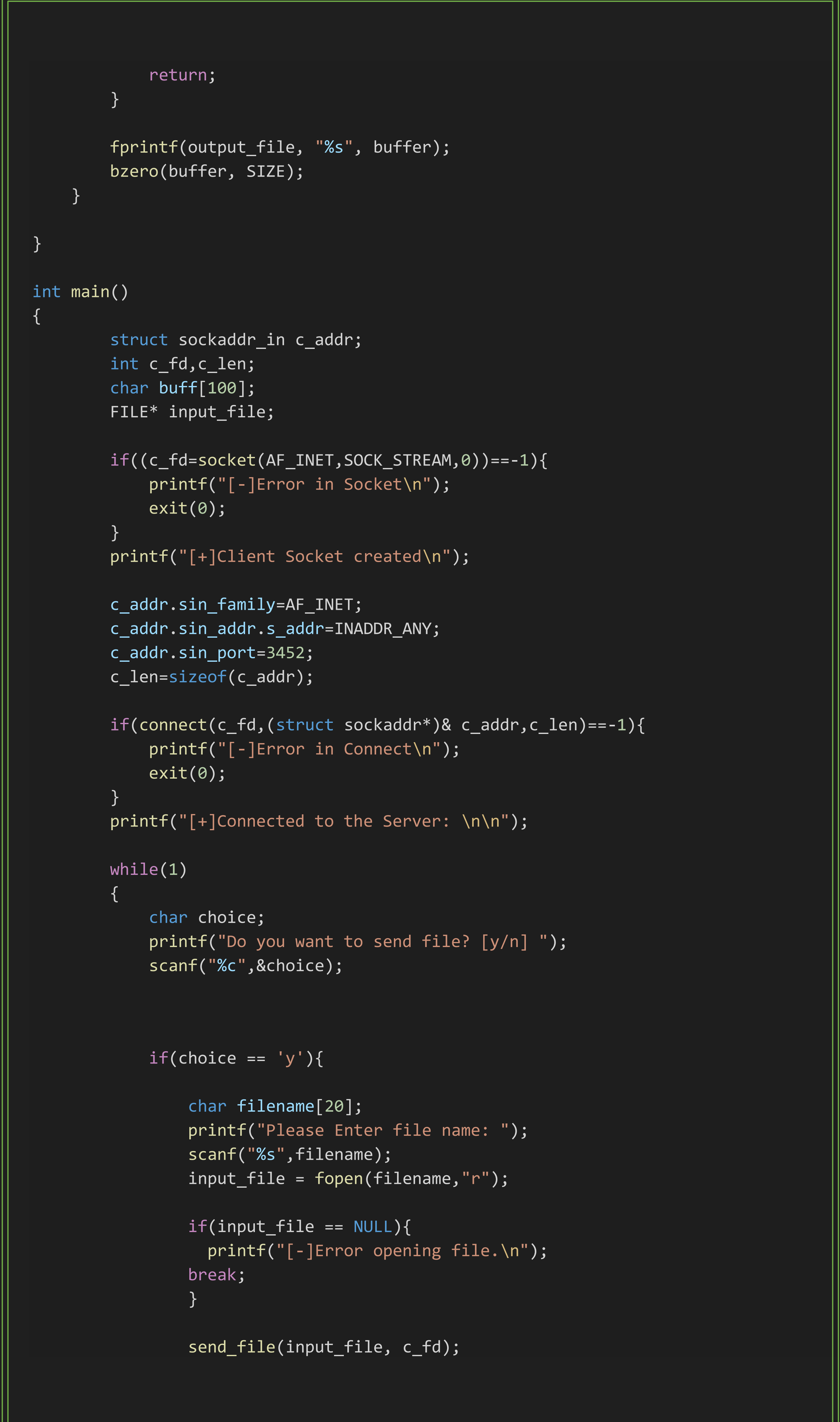
Code Links (from my GITHUB account):

- Client Code
- Server Code

3. Write a program to transfer a text file from the client to server and vice-versa.

Client code

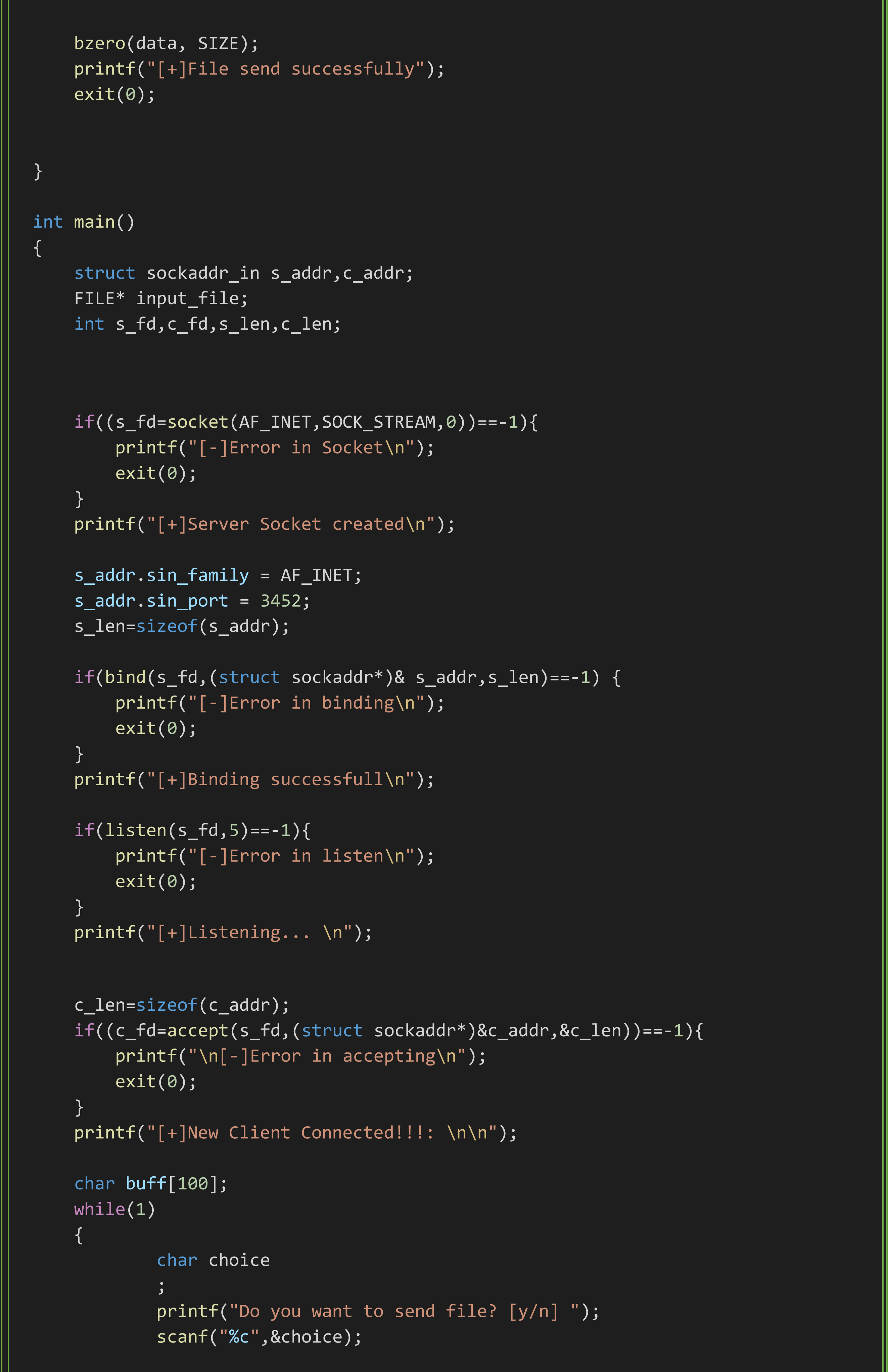
```
#include<stdio.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<string.h>
#include<stdlib.h>

#define SIZE 1024

void send_file(FILE* fp, int socket_fd){
    int n = 0;
    char data[SIZE] = {0};

    char current;
    do{
        current = fgetc(fp);
        if (current != EOF)
            data[n++] = current;
    }while(current !=  EOF);

    if(send(socket_fd,data,SIZE,0) == -1){
        printf("[-]Error sending file");
    }
    bzero(data, SIZE);
    printf("[+]File send successfully");
    exit(0);


}

void write_file(int socket_fd){

    int n;
    FILE* output_file;
    char *filename = "output_file.txt";
    char buffer[SIZE];

    output_file = fopen(filename,"w");
    if(output_file == NULL){
        printf("[-]Error opening file.\n");
    }

    while(1){
        n = recv(socket_fd,buffer,SIZE,0);
        if (n <= 0){
            break;
```



```
            return;
        }

        fprintf(output_file, "%s", buffer);
        bzero(buffer, SIZE);
    }

}

int main()
{
        struct sockaddr_in c_addr;
        int c_fd,c_len;
        char buff[100];
        FILE* input_file;

        if((c_fd=socket(AF_INET,SOCK_STREAM,0))==-1){
            printf("[-]Error in Socket\n");
            exit(0);
        }
        printf("[+]Client Socket created\n");

        c_addr.sin_family=AF_INET;
        c_addr.sin_addr.s_addr=INADDR_ANY;
        c_addr.sin_port=3452;
        c_len=sizeof(c_addr);

        if(connect(c_fd,(struct sockaddr*)& c_addr,c_len)==-1){
            printf("[-]Error in Connect\n");
            exit(0);
        }
        printf("[+]Connected to the Server: \n\n");

        while(1)
        {
            char choice;
            printf("Do you want to send file? [y/n] ");
            scanf("%c",&choice);



            if(choice == 'y'){

                char filename[20];
                printf("Please Enter file name: ");
                scanf("%s",filename);
                input_file = fopen(filename,"r");

                if(input_file == NULL){
                  printf("[-]Error opening file.\n");
                break;
                }

                send_file(input_file, c_fd);
            }

            if(choice == 'n'){
                printf("\n[+]Recieving file Please Wait...\n");
                write_file(c_fd);
                printf("[+]File recieved successfully!!!.");
                exit(0);
            }
        }
        close(c_fd);
        return 0;
}
```

Server code

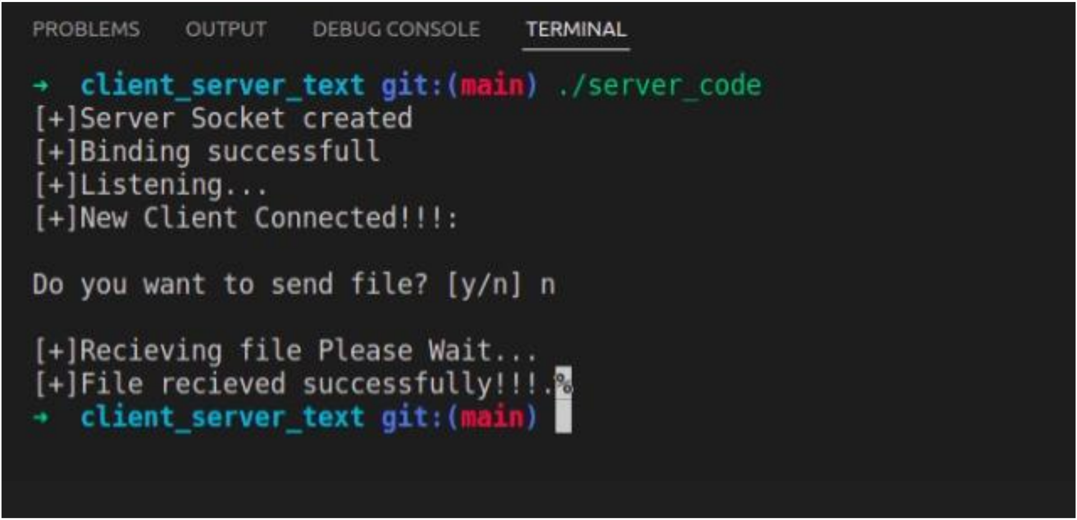
```
#include<stdio.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<stdlib.h>
#include<string.h>

#define SIZE 1024

void write_file(int socket_fd){

    int n;
    FILE* output_file;
    char *filename = "output_file.txt";
    char buffer[SIZE];

    output_file = fopen(filename,"w");
    if(output_file == NULL){
        printf("[-]Error opening file.\n");
    }

    while(1){
        n = recv(socket_fd,buffer,SIZE,0);
        if (n <= 0){
            break;
            return;
        }

        fprintf(output_file, "%s", buffer);
        bzero(buffer, SIZE);
    }

}

void send_file(FILE* fp, int socket_fd){
    int n = 0;
    char data[SIZE] = {0};

    char current;
    do{
        current = fgetc(fp);
        if (current != EOF)
            data[n++] = current;
    }while(current !=  EOF);

    if(send(socket_fd,data,SIZE,0) == -1){
        printf("[-]Error sending file");
    }
```



```c
    bzero(data, SIZE);
    printf("[+]File send successfully");
    exit(0);


}

int main()
{
    struct sockaddr_in s_addr,c_addr;
    FILE* input_file;
    int s_fd,c_fd,s_len,c_len;



    if((s_fd=socket(AF_INET,SOCK_STREAM,0))==-1){
        printf("[-]Error in Socket\n");
        exit(0);
    }
    printf("[+]Server Socket created\n");

    s_addr.sin_family = AF_INET;
    s_addr.sin_port = 3452;
    s_len=sizeof(s_addr);

    if(bind(s_fd,(struct sockaddr*)& s_addr,s_len)==-1) {
        printf("[-]Error in binding\n");
        exit(0);
    }
    printf("[+]Binding successfull\n");

    if(listen(s_fd,5)==-1){
        printf("[-]Error in listen\n");
        exit(0);
    }
    printf("[+]Listening... \n");


    c_len=sizeof(c_addr);
    if((c_fd=accept(s_fd,(struct sockaddr*)&c_addr,&c_len))==-1){
        printf("\n[-]Error in accepting\n");
        exit(0);
    }
    printf("[+]New Client Connected!!!: \n\n");

    char buff[100];
    while(1)
    {
            char choice
            ;
            printf("Do you want to send file? [y/n] ");
            scanf("%c",&choice);
            if(choice == 'y'){

                char filename[20];
                printf("Please Enter file name: ");
                scanf("%s",filename);
                input_file = fopen(filename,"r");

                if(input_file == NULL){
                  printf("[-]Error opening file.\n");
                break;
                }

                send_file(input_file, c_fd);

            }

            if(choice == 'n'){
                printf("\n[+]Recieving file Please Wait...\n");
                write_file(c_fd);
                printf("[+]File recieved successfully!!!.");
                exit(0);
            }

    }
    close(c_fd);
    return 0;
}
```

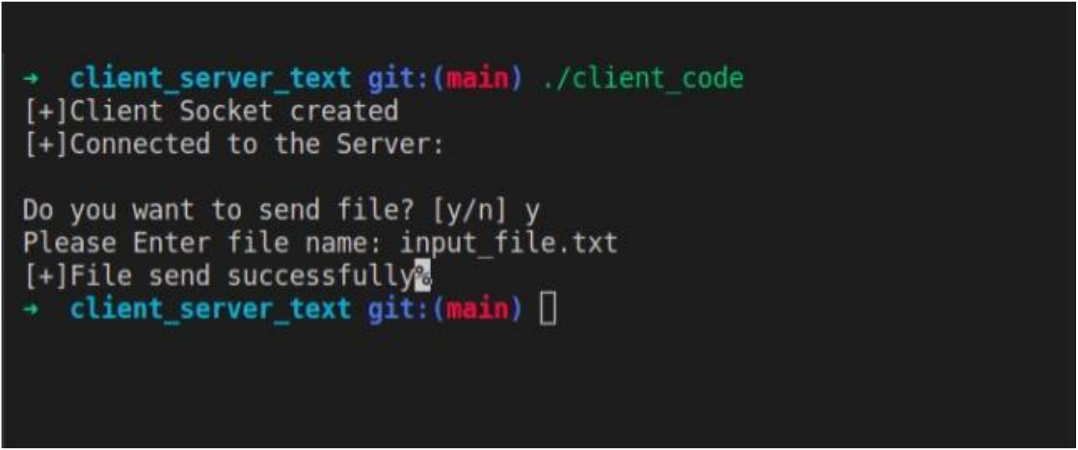
Output:

Server:



```
→  client_server_text git:(main) ./server_code
[+]Server Socket created
[+]Binding successfull
[+]Listening...
[+]New Client Connected!!!:

Do you want to send file? [y/n] n

[+]Recieving file Please Wait...
[+]File recieved successfully!!!.%
→  client_server_text git:(main)
```

Client:



```
→  client_server_text git:(main) ./client_code
[+]Client Socket created
[+]Connected to the Server:

Do you want to send file? [y/n] y
Please Enter file name: input_file.txt
[+]File send successfully%
→  client_server_text git:(main) 
```

Code Links (from my GITHUB account):

- Client Code
- Server Code
- Input File
- Output File

4. A database is created with the following fields: roll no, student name, address, marks1, marks2. The database will be stored in the server and the client will fetch the information of a student by sending the roll no of a particular student. Implement this scenario using client server program.

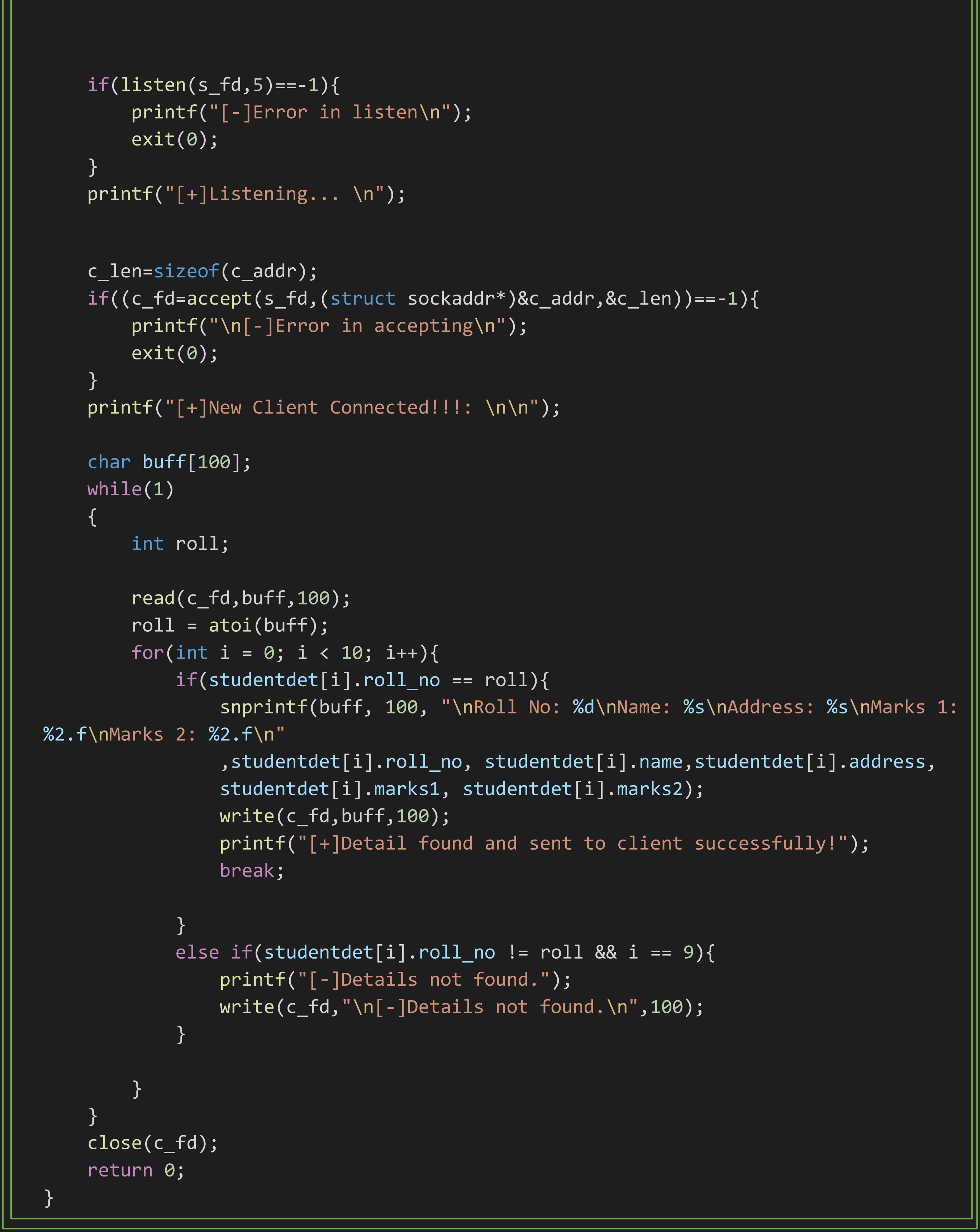
Client code

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

int main()
{
        struct sockaddr_in c_addr;
        int c_fd,c_len;
        char buff[100];

        if((c_fd=socket(AF_INET,SOCK_STREAM,0))==-1){
            printf("[-]Error in Socket\n");
            exit(0);
        }
        printf("[+]Client Socket created\n");

        c_addr.sin_family=AF_INET;
        c_addr.sin_addr.s_addr=INADDR_ANY;
        c_addr.sin_port=3452;
        c_len=sizeof(c_addr);

        if(connect(c_fd,(struct sockaddr*)& c_addr,c_len)==-1) {
            printf("[-]Error in Connect\n");
            exit(0);
        }
        printf("[+]Connected to the Server: \n\n");

        while(1)
        {
            printf("\nEnter Roll No of Student: \n");
            fgets(buff,sizeof(buff),stdin);

            if (strcmp(buff, "Exit\n") == 0){
                printf("[-]Disconneted from Server");
                write(c_fd,"Disconnected\n",100);
                break;
            }
            else{
                write(c_fd,buff,100);
            }
            read(c_fd,buff,100);
            printf("%s",buff);
        }
        close(c_fd);
        return 0;
}
```

Server code

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


struct StudentDetails
{
    int roll_no;
    char* name;
    char* address;
    float marks1;
    float marks2;
};


int main()
{
    struct sockaddr_in s_addr,c_addr;
    int s_fd,c_fd,s_len,c_len;

    struct StudentDetails studentdet[13];

    studentdet[0].roll_no = 1;
    studentdet[0].name = "Biki Paul";
    studentdet[0].address = "Uttarayan,Alipurduar";
    studentdet[0].marks1 = 89.0;
    studentdet[0].marks2 = 99.0;

    studentdet[1].roll_no = 2;
    studentdet[1].name = "Prakash Jha";
    studentdet[1].address = "DS Colony,Alipurduar";
    studentdet[1].marks1 = 81.0;
    studentdet[1].marks2 = 92.0;

    studentdet[2].roll_no = 3;
    studentdet[2].name = "Nabajit Sarmah";
    studentdet[2].address = "DRM,Alipurduar";
    studentdet[2].marks1 = 76.0;
    studentdet[2].marks2 = 87.0;

    studentdet[3].roll_no = 4;
    studentdet[3].name = "Dhruva Boro";
    studentdet[3].address = "DS Colony,Alipurduar";
    studentdet[3].marks1 = 87.0;
    studentdet[3].marks2 = 94.0;

    studentdet[4].roll_no = 5;
```

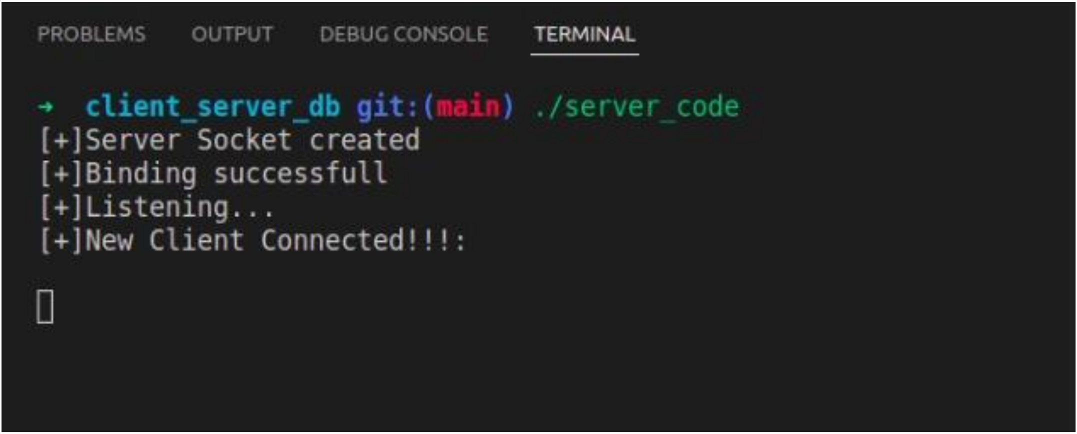
```c
studentdet[4].name = "Dinesh Rao";
studentdet[4].address = "Officers Colony,Alipurduar";
studentdet[4].marks1 = 77.0;
studentdet[4].marks2 = 77.0;

studentdet[5].roll_no = 6;
studentdet[5].name = "Somyajit Dhar";
studentdet[5].address = "Damanpur,Alipurduar";
studentdet[5].marks1 = 89.0;
studentdet[5].marks2 = 91.0;

studentdet[6].roll_no = 7;
studentdet[6].name = "Akash Deb";
studentdet[6].address = "Junction,Alipurduar";
studentdet[6].marks1 = 81.0;
studentdet[6].marks2 = 91.0;

studentdet[7].roll_no = 8;
studentdet[7].name = "Pryobrata Sarkar";
studentdet[7].address = "Nabin Club,Alipurduar";
studentdet[7].marks1 = 86.0;
studentdet[7].marks2 = 90.0;

studentdet[8].roll_no = 9;
studentdet[8].name = "Anuska Sarkar";
studentdet[8].address = "Madhavmore,Alipurduar";
studentdet[8].marks1 = 90.0;
studentdet[8].marks2 = 90.0;

studentdet[9].roll_no = 10;
studentdet[9].name = "Ujjan Sarkar";
studentdet[9].address = "Arabindanagar,Alipurduar";
studentdet[9].marks1 = 89.0;
studentdet[9].marks2 = 92.0;



if((s_fd=socket(AF_INET,SOCK_STREAM,0))==-1){
    printf("[-]Error in Socket\n");
    exit(0);
}
printf("[+]Server Socket created\n");

s_addr.sin_family = AF_INET;
s_addr.sin_port = 3452;
s_len=sizeof(s_addr);

if(bind(s_fd,(struct sockaddr*)& s_addr,s_len)==-1){
    printf("[-]Error in binding\n");
    exit(0);
}
printf("[+]Binding successfull\n");
```



```c
    if(listen(s_fd,5)==-1){
        printf("[-]Error in listen\n");
        exit(0);
    }
    printf("[+]Listening... \n");


    c_len=sizeof(c_addr);
    if((c_fd=accept(s_fd,(struct sockaddr*)&c_addr,&c_len))==-1){
        printf("\n[-]Error in accepting\n");
        exit(0);
    }
    printf("[+]New Client Connected!!!: \n\n");

    char buff[100];
    while(1)
    {
        int roll;

        read(c_fd,buff,100);
        roll = atoi(buff);
        for(int i = 0; i < 10; i++){
            if(studentdet[i].roll_no == roll){
                snprintf(buff, 100, "\nRoll No: %d\nName: %s\nAddress: %s\nMarks 1:
%2.f\nMarks 2: %2.f\n"
                ,studentdet[i].roll_no, studentdet[i].name,studentdet[i].address,
                studentdet[i].marks1, studentdet[i].marks2);
                write(c_fd,buff,100);
                printf("[+]Detail found and sent to client successfully!");
                break;

            }
            else if(studentdet[i].roll_no != roll && i == 9){
                printf("[-]Details not found.");
                write(c_fd,"\n[-]Details not found.\n",100);
            }

        }
    }
    close(c_fd);
    return 0;
}
```

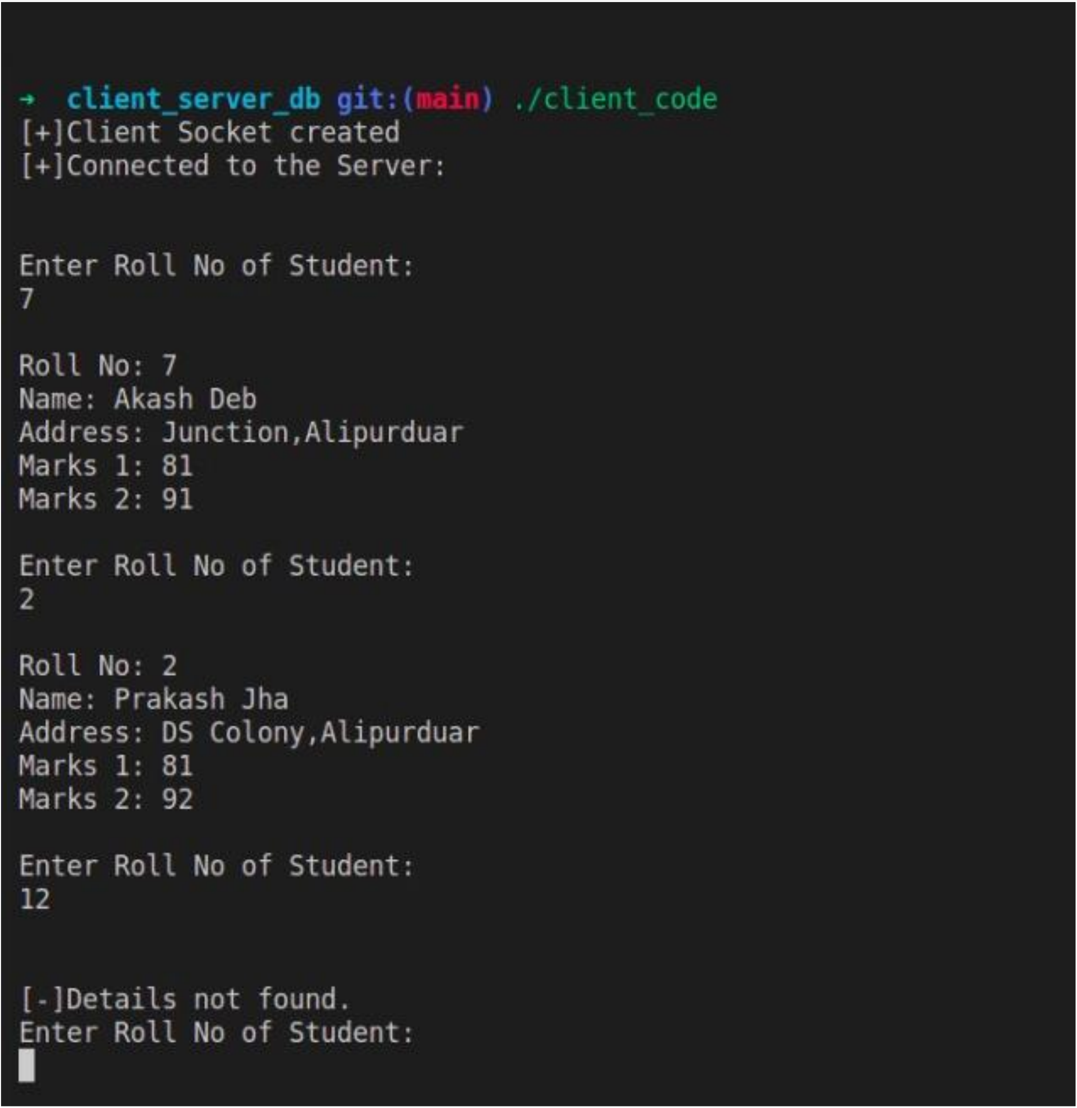
Output:

Server:



```
→  client_server_db git:(main) ./server_code
[+]Server Socket created
[+]Binding successfull
[+]Listening...
[+]New Client Connected!!!:
```

Client:



```
→  client_server_db git:(main) ./client_code
[+]Client Socket created
[+]Connected to the Server:


Enter Roll No of Student:
7

Roll No: 7
Name: Akash Deb
Address: Junction,Alipurduar
Marks 1: 81
Marks 2: 91

Enter Roll No of Student:
2

Roll No: 2
Name: Prakash Jha
Address: DS Colony,Alipurduar
Marks 1: 81
Marks 2: 92

Enter Roll No of Student:
12


[-]Details not found.
Enter Roll No of Student:
```

Code Links (from my GITHUB account):

- Client Code
- Server Code

5. Write a program to implement Stop & Wait ARQ method.

Client code

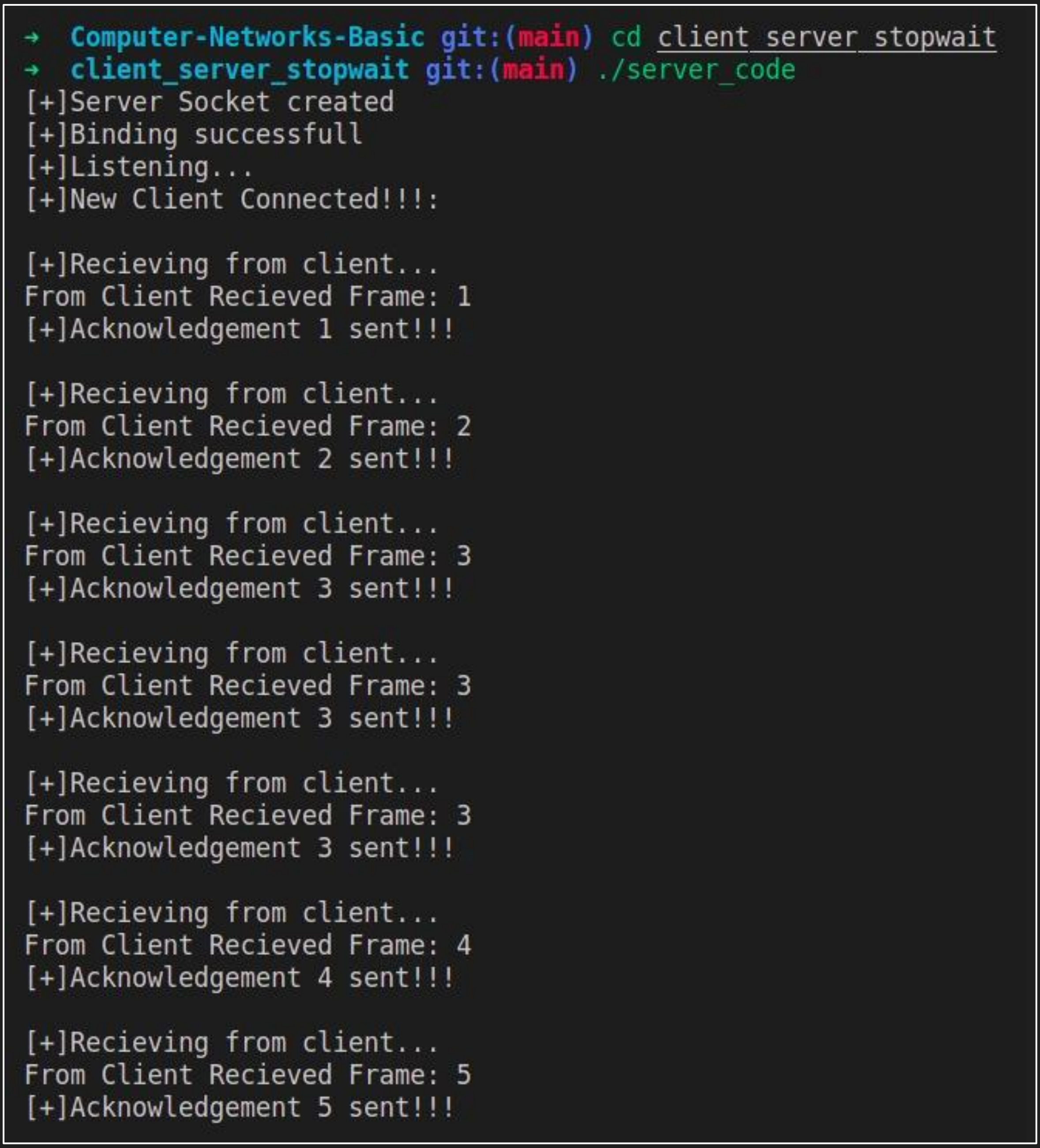
```
#include<stdio.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<string.h>
#include<stdlib.h>

int main()
{
        struct sockaddr_in c_addr;
        int c_fd,c_len;
        char buff[100];
        char buff1[10];

        if((c_fd=socket(AF_INET,SOCK_STREAM,0))==-1){
            printf("[-]Error in Socket\n");
            exit(0);
        }
        printf("[+]Client Socket created\n");

        c_addr.sin_family=AF_INET;
        c_addr.sin_addr.s_addr=INADDR_ANY;
        c_addr.sin_port=3452;
        c_len=sizeof(c_addr);

        if(connect(c_fd,(struct sockaddr*)& c_addr,c_len)==-1)  {
            printf("[-]Error in Connect\n");
            exit(0);
        }
        printf("[+]Connected to the Server: \n\n");

        while(1)
        {
            srand(time(0));

            printf("Enter your Frames Mr.Client: ");
            fgets(buff1,sizeof(buff1),stdin);


            for(int i = 0; i < sizeof(buff1); i++){
                    if(isdigit(buff1[i])){
                        buff[0] = buff1[i];
                        write(c_fd,buff,100);
                        printf("\n[+]Frame %c send successfully!!!\n",buff[0]);
                        printf("\n[.]Waiting for acknowledgement...\n");
                        int randomnumber;
                        randomnumber = rand() % 10 + 1;
                        read(c_fd,buff,100);


                        while(randomnumber > 7){
                            printf("\n[-]Timeout!!!\n");
                            printf("\n[-]Delayed Acknowledgement\n");
                            printf("\n[-]Sending Frame  %c again\n",buff1[i]);
                            bzero(buff,sizeof(buff));
                            buff[0] = buff1[i];
                            write(c_fd,buff,100);
                            printf("\n[+]Frame %c send successfully!!!\n",buff1[i]);
                            printf("\n[.]Waiting for acknowledgement...\n");
                            read(c_fd,buff,100);
                            randomnumber = rand() % 10 + 1;

                        }

                            printf("\n[+]Acknowledgement: %c\n",buff[0]);
                            bzero(buff,sizeof(buff));

                    }
                }

        }
        close(c_fd);
        return 0;
}
```

Server code

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

int main()
{
    struct sockaddr_in s_addr,c_addr;

    int s_fd,c_fd,s_len,c_len;



    if((s_fd=socket(AF_INET,SOCK_STREAM,0))==-1)    {
        printf("[-]Error in Socket\n");
        exit(0);
    }
    printf("[+]Server Socket created\n");

    s_addr.sin_family = AF_INET;
    s_addr.sin_port = 3452;
    s_len=sizeof(s_addr);

    if(bind(s_fd,(struct sockaddr*)& s_addr,s_len)==-1) {
        printf("[-]Error in binding\n");
        exit(0);
    }
    printf("[+]Binding successfull\n");

    if(listen(s_fd,5)==-1)  {
        printf("[-]Error in listen\n");
        exit(0);
    }
    printf("[+]Listening... \n");


    c_len=sizeof(c_addr);
    if((c_fd=accept(s_fd,(struct sockaddr*)&c_addr,&c_len))==-1){
        printf("\n[-]Error in accepting\n");
        exit(0);
    }
    printf("[+]New Client Connected!!!: \n\n");

    char buff[100];
    while(1)
    {

        printf("[+]Recieving from client...");
        read(c_fd,buff,100);
        printf("\nFrom Client Recieved Frame: %s\n",buff);
        printf("[+]Acknowledgement %c sent!!!\n\n",buff[0]);
        write(c_fd,buff,100);


    }
    close(c_fd);
    return 0;
}
```

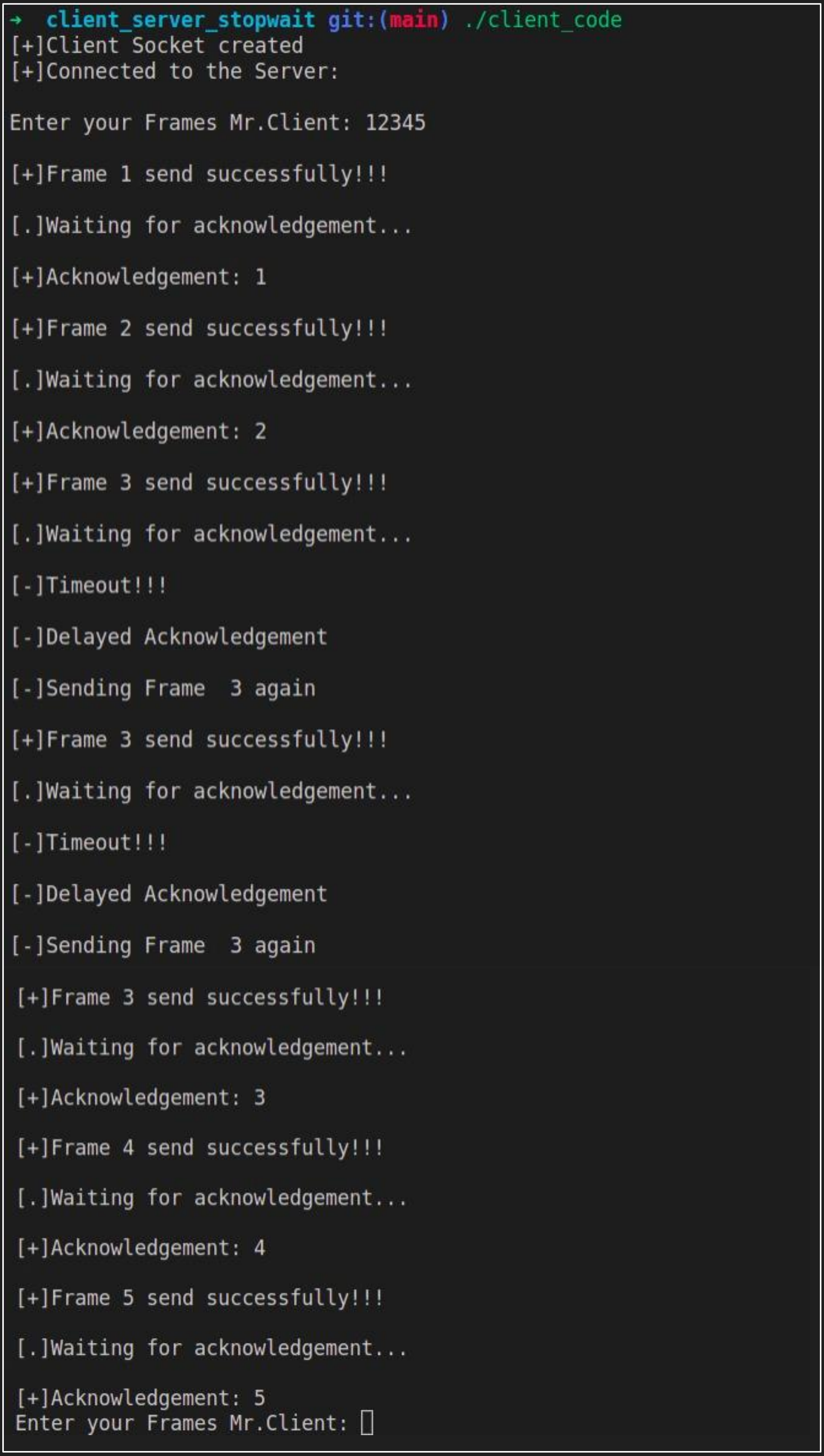
Output:

Server:



```
➜  Computer-Networks-Basic git:(main) cd client_server_stopwait
➜  client_server_stopwait git:(main) ./server_code
[+]Server Socket created
[+]Binding successfull
[+]Listening...
[+]New Client Connected!!!:

[+]Recieving from client...
From Client Recieved Frame: 1
[+]Acknowledgement 1 sent!!!

[+]Recieving from client...
From Client Recieved Frame: 2
[+]Acknowledgement 2 sent!!!

[+]Recieving from client...
From Client Recieved Frame: 3
[+]Acknowledgement 3 sent!!!

[+]Recieving from client...
From Client Recieved Frame: 3
[+]Acknowledgement 3 sent!!!

[+]Recieving from client...
From Client Recieved Frame: 3
[+]Acknowledgement 3 sent!!!

[+]Recieving from client...
From Client Recieved Frame: 4
[+]Acknowledgement 4 sent!!!

[+]Recieving from client...
From Client Recieved Frame: 5
[+]Acknowledgement 5 sent!!!
```

Client:



```
→  client_server_stopwait git:(main) ./client_code
[+]Client Socket created
[+]Connected to the Server:

Enter your Frames Mr.Client: 12345

[+]Frame 1 send successfully!!!

[.]Waiting for acknowledgement...

[+]Acknowledgement: 1

[+]Frame 2 send successfully!!!

[.]Waiting for acknowledgement...

[+]Acknowledgement: 2

[+]Frame 3 send successfully!!!

[.]Waiting for acknowledgement...

[-]Timeout!!!

[-]Delayed Acknowledgement

[-]Sending Frame  3 again

[+]Frame 3 send successfully!!!

[.]Waiting for acknowledgement...

[-]Timeout!!!

[-]Delayed Acknowledgement

[-]Sending Frame  3 again

[+]Frame 3 send successfully!!!

[.]Waiting for acknowledgement...

[+]Acknowledgement: 3

[+]Frame 4 send successfully!!!

[.]Waiting for acknowledgement...

[+]Acknowledgement: 4

[+]Frame 5 send successfully!!!

[.]Waiting for acknowledgement...

[+]Acknowledgement: 5
Enter your Frames Mr.Client: 
```

Code Links (from my GITHUB account):

- Client Code
- Server Code

6. Write a program to implement Go-Back-N ARQ method.

Client code

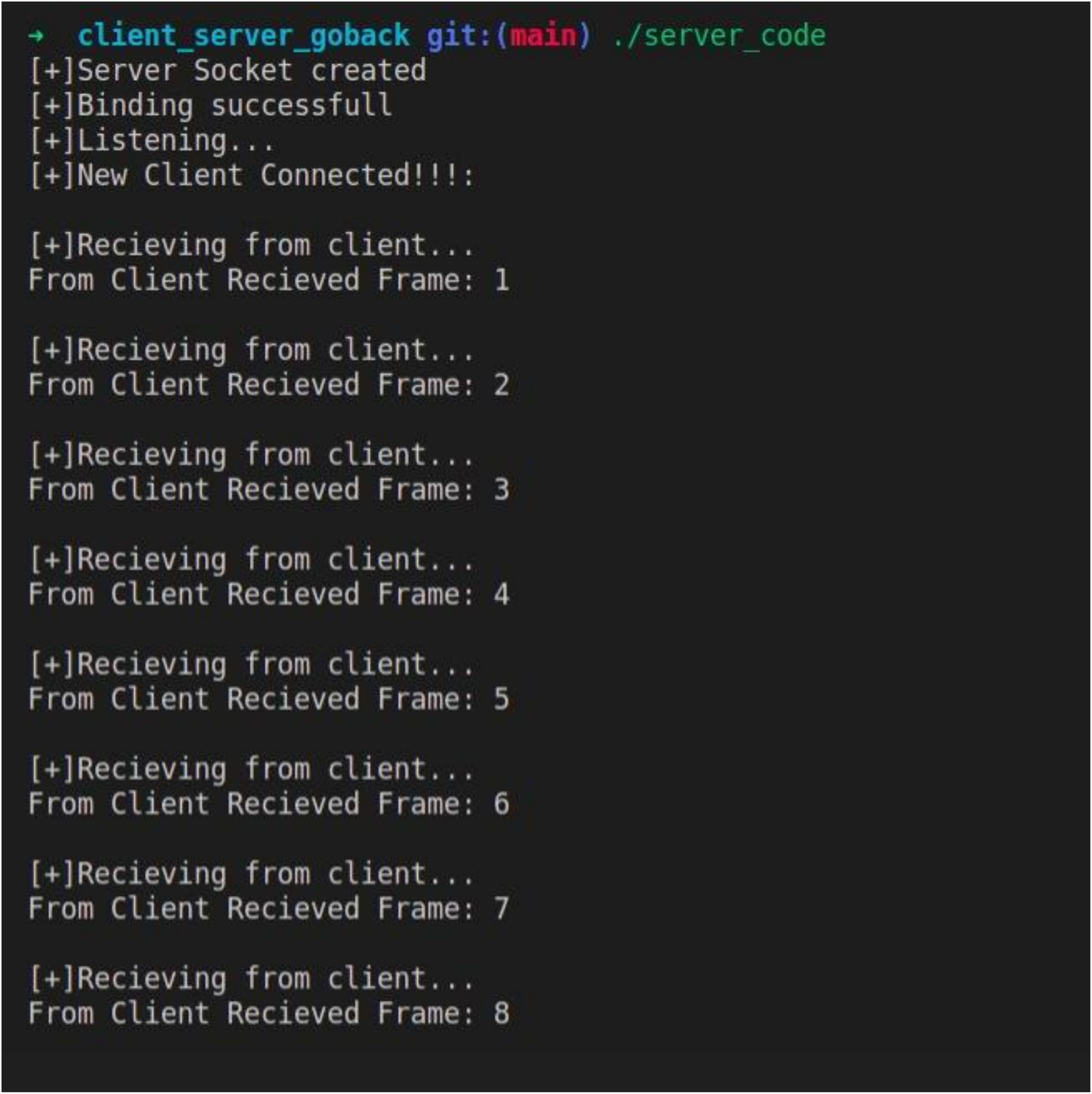
```
#include<stdio.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<string.h>
#include<stdlib.h>

int main()
{
        struct sockaddr_in c_addr;
        int c_fd,c_len;
        char buff[100];
        char buff1[2];

        if((c_fd=socket(AF_INET,SOCK_STREAM,0))==-1){
            printf("[-]Error in Socket\n");
            exit(0);
        }
        printf("[+]Client Socket created\n");

        c_addr.sin_family=AF_INET;
        c_addr.sin_addr.s_addr=INADDR_ANY;
        c_addr.sin_port=3452;
        c_len=sizeof(c_addr);

        if(connect(c_fd,(struct sockaddr*)& c_addr,c_len)==-1) {
            printf("[-]Error in Connect\n");
            exit(0);
        }
        printf("[+]Connected to the Server: \n\n");

        while(1)
        {
            srand(time(0));
            int framesize;
            int windowsize;
            printf("Enter Frames Size Mr.Client: ");
            scanf("%d",&framesize);
            printf("Enter Window Size Mr.Client: ");
            scanf("%d",&windowsize);
            int i = 1;
            while(i <= framesize){

                int z = 0;
                for(int k = i; k < i+windowsize && k <= framesize; k++){
                    sprintf(buff1,"%d",k);
                    buff[0] = buff1[0];
                    write(c_fd,buff,100);
                    printf("\n[+]Frame %c send successfully!!!\n",buff[0]);
                }

                for(int k = i; k < i+windowsize && k <= framesize; k++){
                    int randomnumber;
                    randomnumber = rand() % 10 + 1;

                    if(randomnumber <= 8){
                        // read(c_fd,buff,100);
                        printf("\n[+]Acknowledgement: %d\n",k);
                        // bzero(buff,sizeof(buff));
                        z++;
                    }
                    else
                    {
                        printf("\n[-]Timeout!!!\n");
                        printf("\n[-]Delayed Acknowledgement\n");
                        printf("\n[-]Frame  %d not recieved\n",k);
                        printf("\n[+]Retrensmitting Window\n");
                        break;
                    }
                }

                printf("\n");
                i = i + z;
            }

        }
        close(c_fd);
        return 0;
}
```

Server code

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

int main()
{
    struct sockaddr_in s_addr,c_addr;

    int s_fd,c_fd,s_len,c_len;



    if((s_fd=socket(AF_INET,SOCK_STREAM,0))==-1){
        printf("[-]Error in Socket\n");
        exit(0);
    }
    printf("[+]Server Socket created\n");

    s_addr.sin_family = AF_INET;
    s_addr.sin_port = 3452;
    s_len=sizeof(s_addr);

    if(bind(s_fd,(struct sockaddr*)& s_addr,s_len)==-1) {
        printf("[-]Error in binding\n");
        exit(0);
    }
    printf("[+]Binding successfull\n");

    if(listen(s_fd,5)==-1) {
        printf("[-]Error in listen\n");
        exit(0);
    }
    printf("[+]Listening... \n");


    c_len=sizeof(c_addr);
    if((c_fd=accept(s_fd,(struct sockaddr*)&c_addr,&c_len))==-1){
        printf("\n[-]Error in accepting\n");
        exit(0);
    }
    printf("[+]New Client Connected!!!: \n\n");

    char buff[100];
    while(1)
    {

        printf("[+]Recieving from client...");
        read(c_fd,buff,100);
        printf("\nFrom Client Recieved Frame: %s\n\n",buff);
        // write(c_fd,buff,100);


    }
    close(c_fd);
    return 0;
}
```

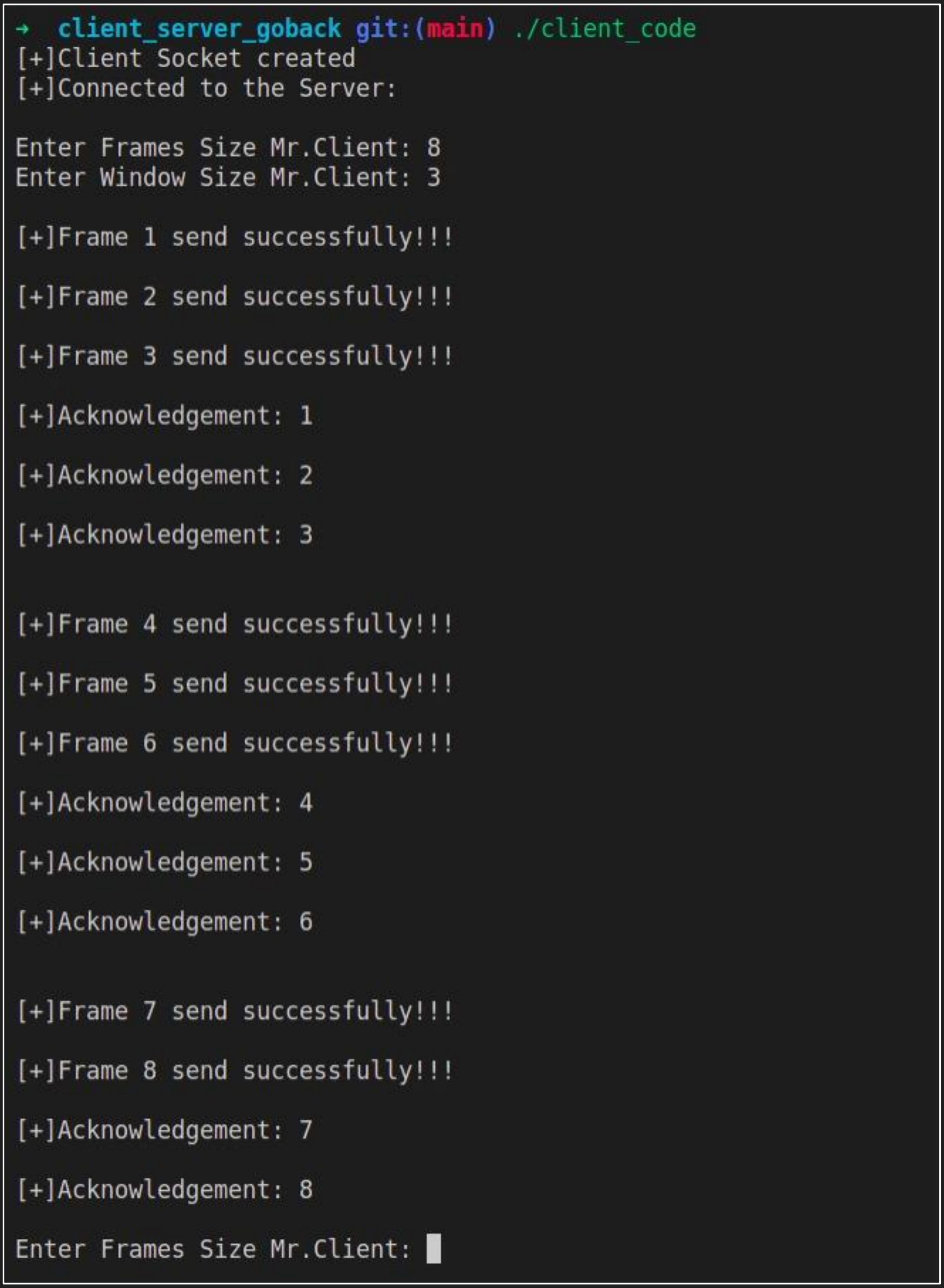
Output:

Server:



```
➜  client_server_goback git:(main) ./server_code
[+]Server Socket created
[+]Binding successfull
[+]Listening...
[+]New Client Connected!!!:

[+]Recieving from client...
From Client Recieved Frame: 1

[+]Recieving from client...
From Client Recieved Frame: 2

[+]Recieving from client...
From Client Recieved Frame: 3

[+]Recieving from client...
From Client Recieved Frame: 4

[+]Recieving from client...
From Client Recieved Frame: 5

[+]Recieving from client...
From Client Recieved Frame: 6

[+]Recieving from client...
From Client Recieved Frame: 7

[+]Recieving from client...
From Client Recieved Frame: 8
```

Client:



```
➜  client_server_goback git:(main) ./client_code
[+]Client Socket created
[+]Connected to the Server:

Enter Frames Size Mr.Client: 8
Enter Window Size Mr.Client: 3

[+]Frame 1 send successfully!!!

[+]Frame 2 send successfully!!!

[+]Frame 3 send successfully!!!

[+]Acknowledgement: 1

[+]Acknowledgement: 2

[+]Acknowledgement: 3


[+]Frame 4 send successfully!!!

[+]Frame 5 send successfully!!!

[+]Frame 6 send successfully!!!

[+]Acknowledgement: 4

[+]Acknowledgement: 5

[+]Acknowledgement: 6


[+]Frame 7 send successfully!!!

[+]Frame 8 send successfully!!!

[+]Acknowledgement: 7

[+]Acknowledgement: 8

Enter Frames Size Mr.Client: 
```

Code Links (from my GITHUB account):

- Client Code
- Server Code

7. Write a program to implement Selective Repeat ARQ method.

Client code

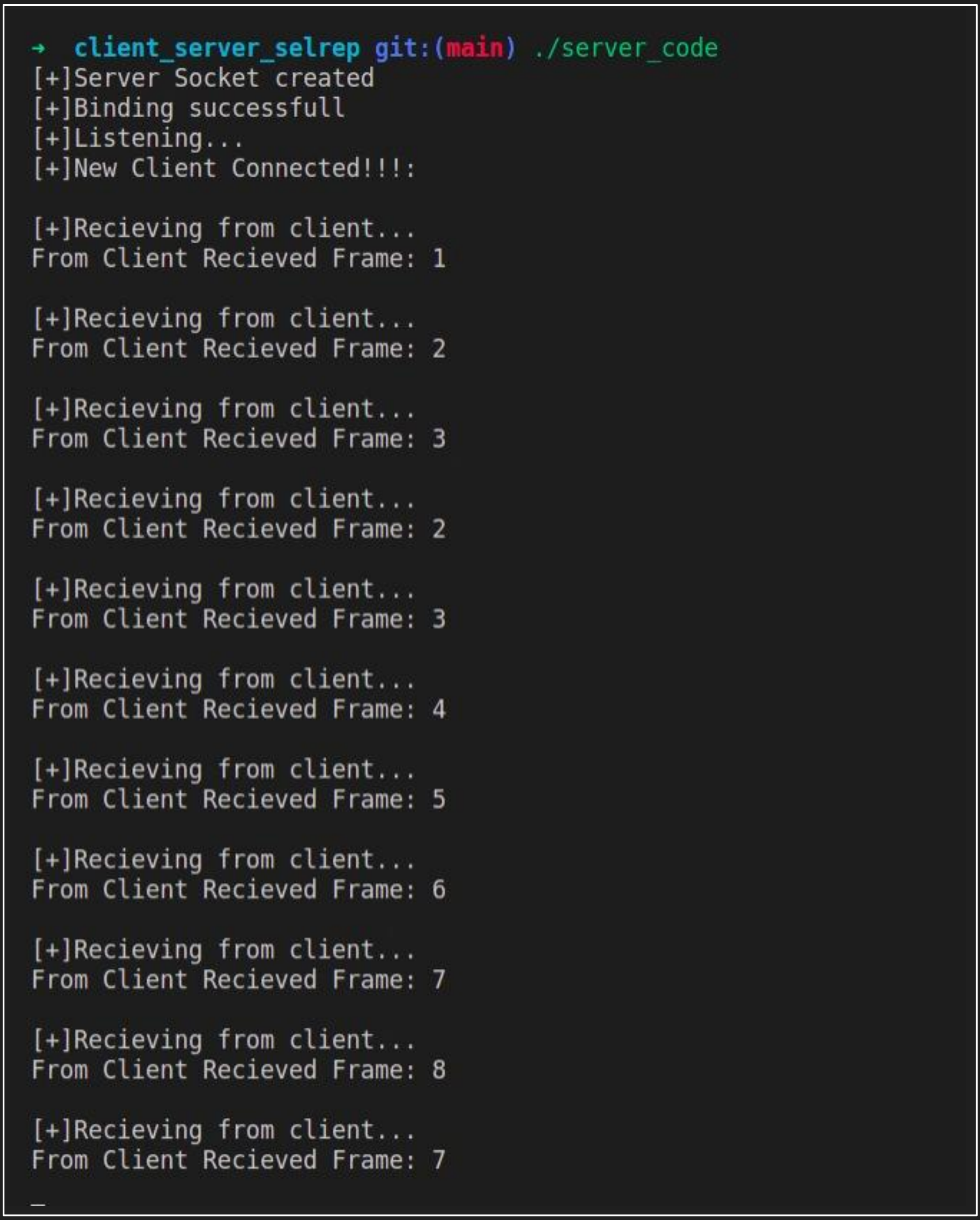
```
#include<stdio.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<string.h>
#include<stdlib.h>

int main()
{
        struct sockaddr_in c_addr;
        int c_fd,c_len;
        char buff[100];
        char buff1[2];

        if((c_fd=socket(AF_INET,SOCK_STREAM,0))==-1){
            printf("[-]Error in Socket\n");
            exit(0);
        }
        printf("[+]Client Socket created\n");

        c_addr.sin_family=AF_INET;
        c_addr.sin_addr.s_addr=INADDR_ANY;
        c_addr.sin_port=4000;
        c_len=sizeof(c_addr);

        if(connect(c_fd,(struct sockaddr*)& c_addr,c_len)==-1) {
            printf("[-]Error in Connect\n");
            exit(0);
        }
        printf("[+]Connected to the Server: \n\n");

        while(1)
        {
            srand(time(0));
            int framesize;
            int windowsize;
            printf("Enter Frames Size Mr.Client: ");
            scanf("%d",&framesize);
            printf("Enter Window Size Mr.Client: ");
            scanf("%d",&windowsize);

            int i = 1;
            while(i <= framesize){

                int z = 0;
                for(int k = i; k < i+windowsize && k <= framesize; k++){
                    sprintf(buff1,"%d",k);
                    buff[0] = buff1[0];
                    write(c_fd,buff,100);
                    printf("\n[+]Frame %c send successfully!!!\n",buff[0]);
                }

                for(int k = i; k < i+windowsize && k <= framesize; k++){
                    int randomnumber;
                    randomnumber = rand() % 10 + 1;

                    if(randomnumber < 7){
                        // read(c_fd,buff,100);
                        printf("\n[+]Acknowledgement: %d\n",k);
                        // bzero(buff,sizeof(buff));
                        z++;
                    }
                    else
                    {
                        printf("\n[-]NAK (Timeout) %d\n",k);
                        printf("\n[-]Frame %d sent again successfully!!!\n",k);
                        sprintf(buff1,"%d",k);
                        buff[0] = buff1[0];
                        write(c_fd,buff,100);
                        printf("\n[+]Acknowledgement: %d\n",k);
                        z++;
                    }
                }

                printf("\n");
                i = i + z;
            }

        }
        close(c_fd);
        return 0;
}
```

Server code

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

int main()
{
    struct sockaddr_in s_addr,c_addr;

    int s_fd,c_fd,s_len,c_len;



    if((s_fd=socket(AF_INET,SOCK_STREAM,0))==-1){
        printf("[-]Error in Socket\n");
        exit(0);
    }
    printf("[+]Server Socket created\n");

    s_addr.sin_family = AF_INET;
    s_addr.sin_port = 4000;
    s_len=sizeof(s_addr);

    if(bind(s_fd,(struct sockaddr*)& s_addr,s_len)==-1){
        printf("[-]Error in binding\n");
        exit(0);
    }
    printf("[+]Binding successfull\n");

    if(listen(s_fd,5)==-1){
        printf("[-]Error in listen\n");
        exit(0);
    }
    printf("[+]Listening... \n");


    c_len=sizeof(c_addr);
    if((c_fd=accept(s_fd,(struct sockaddr*)&c_addr,&c_len))==-1){
        printf("\n[-]Error in accepting\n");
        exit(0);
    }
    printf("[+]New Client Connected!!!: \n\n");

    char buff[100];
    while(1)
    {

        printf("[+]Recieving from client...");
        read(c_fd,buff,100);
        printf("\nFrom Client Recieved Frame: %s\n\n",buff);
        // write(c_fd,buff,100);


    }
    close(c_fd);
    return 0;
}
```

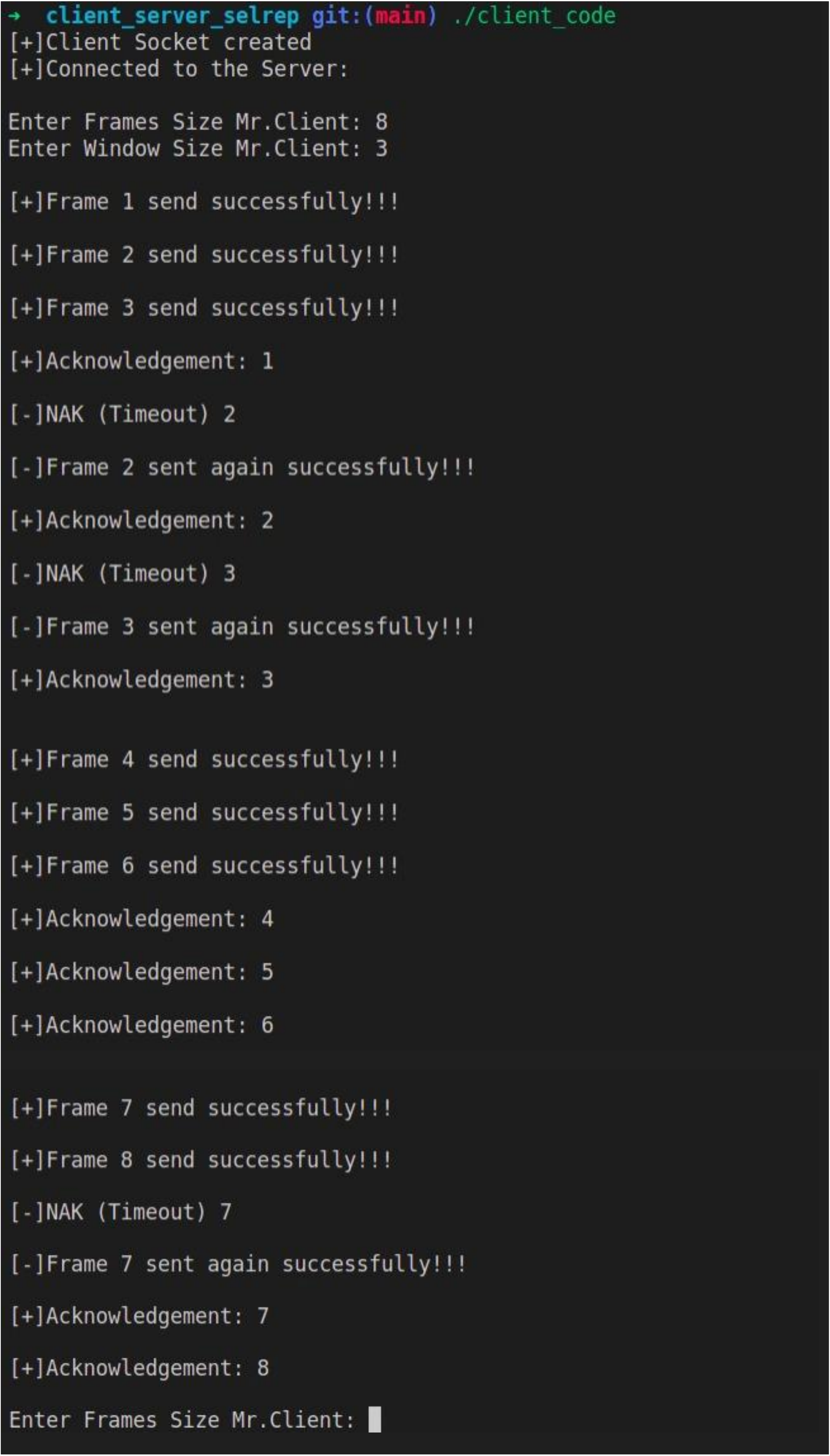
Output:

Server:



```
➜  client_server_selrep git:(main) ./server_code
[+]Server Socket created
[+]Binding successfull
[+]Listening...
[+]New Client Connected!!!:

[+]Recieving from client...
From Client Recieved Frame: 1

[+]Recieving from client...
From Client Recieved Frame: 2

[+]Recieving from client...
From Client Recieved Frame: 3

[+]Recieving from client...
From Client Recieved Frame: 2

[+]Recieving from client...
From Client Recieved Frame: 3

[+]Recieving from client...
From Client Recieved Frame: 4

[+]Recieving from client...
From Client Recieved Frame: 5

[+]Recieving from client...
From Client Recieved Frame: 6

[+]Recieving from client...
From Client Recieved Frame: 7

[+]Recieving from client...
From Client Recieved Frame: 8

[+]Recieving from client...
From Client Recieved Frame: 7

_
```

Client:



```
→  client_server_selrep git:(main) ./client_code
[+]Client Socket created
[+]Connected to the Server:

Enter Frames Size Mr.Client: 8
Enter Window Size Mr.Client: 3

[+]Frame 1 send successfully!!!

[+]Frame 2 send successfully!!!

[+]Frame 3 send successfully!!!

[+]Acknowledgement: 1

[-]NAK (Timeout) 2

[-]Frame 2 sent again successfully!!!

[+]Acknowledgement: 2

[-]NAK (Timeout) 3

[-]Frame 3 sent again successfully!!!

[+]Acknowledgement: 3


[+]Frame 4 send successfully!!!

[+]Frame 5 send successfully!!!

[+]Frame 6 send successfully!!!

[+]Acknowledgement: 4

[+]Acknowledgement: 5

[+]Acknowledgement: 6


[+]Frame 7 send successfully!!!

[+]Frame 8 send successfully!!!

[-]NAK (Timeout) 7

[-]Frame 7 sent again successfully!!!

[+]Acknowledgement: 7

[+]Acknowledgement: 8

Enter Frames Size Mr.Client: 
```

Code Links (from my GITHUB account):

- Client Code
- Server Code