Assignment 3 TCP-RUDP
Computer Network

Program: MScIT Sem-2

Group ID : 28

Student Name	Student ID
Dev Adnani	202212012
Saif Saiyed	202212083

202212012

1: Time Control

Client.c

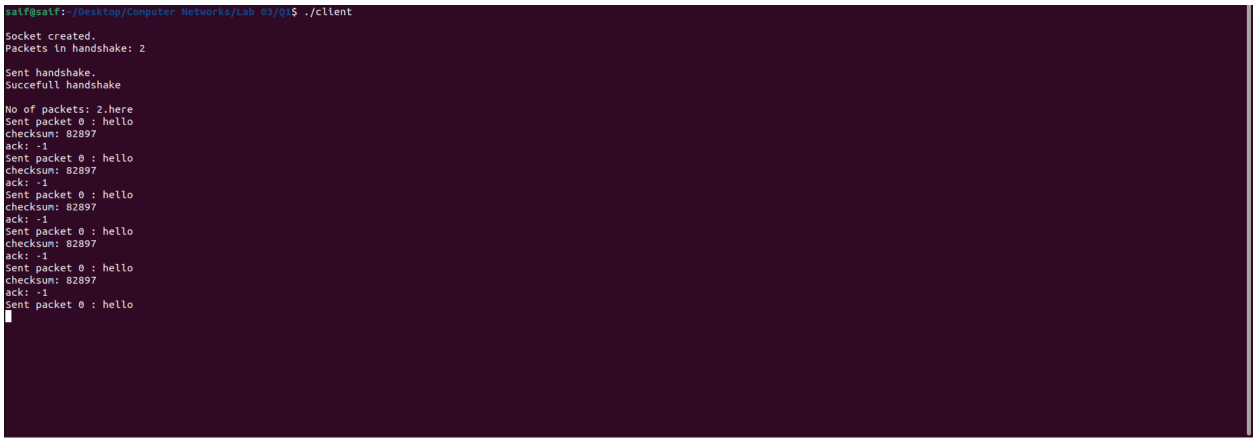
```
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdio.h>

#define PORT 5100


struct control_pkt{
    int sr_no;
    int no_of_pkt;
    int start_sr_no;
    int end_sr_no;
    int checksum;
};


struct data_pkt{
    int sr_no;
    char msg[100];
    long checkSum;
};

struct ack_pkt{
    int ack;
    int sr_no;
    long checkSum;
};


int noOfPck(int size){
    int req = size/100;
    double reqD = size/(double)100;
    if(reqD > req) req++;
    return req;
}
```

```
long createSumControl(struct control_pkt * c_pkt){
    long sum=0;
    sum+=c_pkt->end_sr_no;
    sum+=c_pkt->no_of_pkt;
    sum+=c_pkt->sr_no;
    sum+=c_pkt->start_sr_no;
    return sum;
}

int checkSumControl(struct control_pkt * c_pkt){
    long check = createSumControl(c_pkt);
    if(c_pkt->checksum == check) return 1;
    return 0;
}

long createSumACK(struct ack_pkt* ack){
    return ack->ack+ack->sr_no;
}

int checkSumACK(struct ack_pkt * ack){
    long ackSum = createSumACK(ack);
    if(ackSum == ack->checkSum) return 1;
    return 0;
}

long createSum(struct data_pkt* data){
    long msgSum=0;

    for(int i=0;i<100;i+=2){
        msgSum += ((data->msg[i]<<8) + data->msg[i+1]);
    }

    msgSum += data->sr_no;

    return msgSum;
}

int checkSum(struct data_pkt* data){
    long check = createSum(data);

    if(data->checkSum == check)return 1;
    return 0;
}
```

```
int shakeHand(int sock_fd,struct sockaddr_in server_addr,int sock_length,int size,
   struct control_pkt* c_pkt){

   int pkts = noOfPck(size);

   printf("\nPackets in handshake: %d\n",pkts);

   c_pkt->end_sr_no = pkts;
   c_pkt->sr_no = 0;
   c_pkt->no_of_pkt = pkts;
   c_pkt->start_sr_no=1;
   c_pkt->checksum = createSumControl(c_pkt);

   //ack
   struct ack_pkt ackPkt;
   struct ack_pkt * ack = &ackPkt;

   ack->ack=-1;


   while(ack->ack != 1){
         if(sendto(sock_fd,c_pkt,sizeof(*c_pkt),0,
                  (struct sockaddr *)&server_addr,sock_length)<0){
                        printf("\nError while sending message to server.\n");
                        return -1;
         }
         printf("\nSent handshake.");

         struct timeval t;
         t.tv_sec=5;
         fd_set socks;
         FD_ZERO(&socks);
         FD_SET(sock_fd,&socks);

         if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
                  recvfrom(sock_fd,ack,sizeof(*ack),0,
                  (struct sockaddr *)&server_addr,&sock_length)<0){
                        printf("\nError in delay recv function.");
         }

   }
   return 1;

}
```

```
int comunicate(int sock_fd,struct sockaddr_in server_addr,int sock_length){
    //msg
    struct data_pkt dataSend;
    struct data_pkt * data1 = &dataSend;
    memset(data1->msg,0,100);

    printf("\nEnter msg: ");

    fgets(data1->msg,100,stdin);
    data1->sr_no=1;

    data1->checkSum = createSum(data1);


    if(sendto(sock_fd,data1,sizeof(*data1),0,
            (struct sockaddr *)&server_addr,sock_length)<0){
                    printf("\nError while sending message to server.");
                    return -1;
    }
    printf("\nSent: %s",data1->msg);


    struct ack_pkt dataRecv;
    struct ack_pkt * ack = &dataRecv;
    ack->ack=-1;

    struct timeval t;
    t.tv_sec=5;
    fd_set socks;
    FD_ZERO(&socks);
    FD_SET(sock_fd,&socks);

    if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
            recvfrom(sock_fd,ack,sizeof(*ack),0,
            (struct sockaddr *)&server_addr,&sock_length)<0){
                    printf("\nError in delay recv function.");

    }

    if(ack->ack==-1){
            printf("\nNo ack recved");
    }
    else{
            printf("\nAck recved");
    }
    return 1;
}

int comunicatePkt(int sock_fd,struct sockaddr_in server_addr,int sock_length,char *
msg,int sr_no){

    struct data_pkt dataSend;
    struct data_pkt * data = &dataSend;
    memset(data->msg,0,100);


    struct ack_pkt ackPkt;
    struct ack_pkt * ack = &ackPkt;

    ack->ack=-1;

    int i=0;
    while(*msg != '\0'){
        data->msg[i]=*msg;
        msg++;i++;
    }

    data->sr_no = sr_no;
    data->checkSum = createSum(data);

    if(sendto(sock_fd,data,sizeof(*data),0,
        (struct sockaddr *)&server_addr,sock_length)<0){
                printf("\nError while sending message to server");
                return -1;
    }

    printf("\nSent packet %d : %s ",sr_no,data->msg);
    printf("\nchecksum: %ld",data->checkSum);

    struct timeval t;
    t.tv_sec=5;
    fd_set socks;
    FD_ZERO(&socks);
    FD_SET(sock_fd,&socks);
    if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
        recvfrom(sock_fd,ack,sizeof(*ack),0,
        (struct sockaddr *)&server_addr,&sock_length)<0){
                printf("\nError in delay recv function.");
    }
    printf("\nack: %d",ack->ack);
    return ack->ack;

}


int main(){
    int sock_fd;
    struct sockaddr_in server_addr;
    int sock_length = sizeof(server_addr);

    sock_fd = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);

    if(sock_fd < 0){
            printf("\nError while creating socket.");
            return -1;
    }
    printf("\nSocket created.");

    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    server_addr.sin_port = htons(PORT);
    server_addr.sin_family = AF_INET;

    char msg[200];
    int size = sizeof(msg)/sizeof(msg[0]);

    struct control_pkt pkt;

    int handshakeResult = shakeHand(sock_fd,server_addr,sock_length,size,&pkt);
    while(handshakeResult==-1){
            handshakeResult=shakeHand(sock_fd,server_addr,sock_length,size,&pkt);
    }

    printf("\nSuccefull handshake\n");
    printf("\nNo of packets: %d.",pkt.no_of_pkt);

    char *pkt_datas[2];
    char * f = "hello";
    char * s = "world";
    printf("here");
    pkt_datas[0]=f;
    pkt_datas[1]=s;
    for(int i=0;i<pkt.no_of_pkt;i++){
        int result = comunicatePkt(sock_fd,server_addr,sock_length,pkt_datas[i],i);
        while(result == -1){
            result = comunicatePkt(sock_fd,server_addr,sock_length,pkt_datas[i],i);
        }
    }

    return 0;
}
```

Client Screenshot :



```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q1$ ./client

Socket created.
Packets in handshake: 2

Sent handshake.
Succefull handshake

No of packets: 2.here
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
```

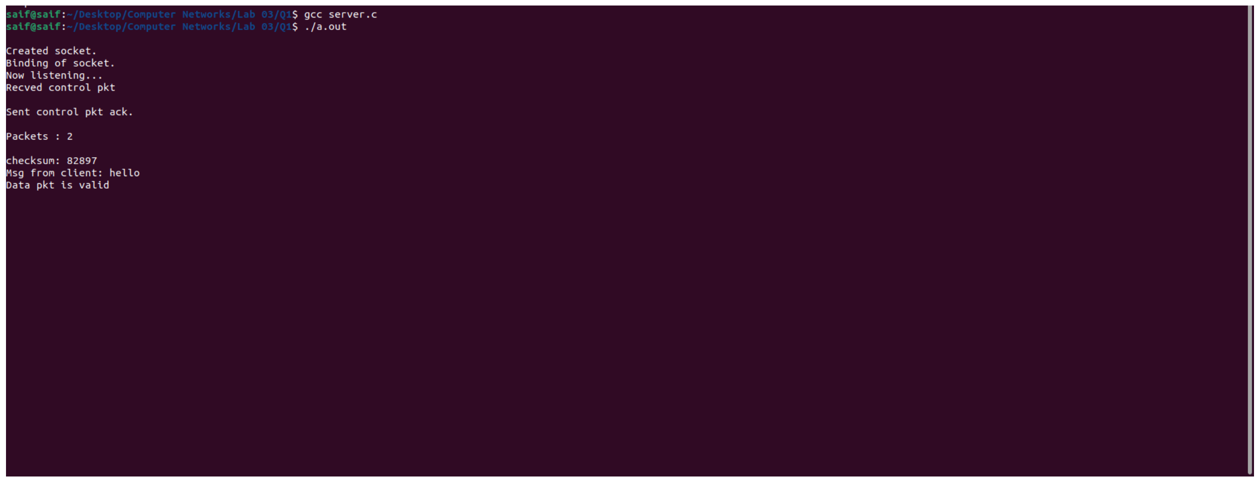
Server.c

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

#define PORT 5100

struct control_pkt{
    int sr_no;
    int no_of_pkt;
    int start_sr_no;
    int end_sr_no;
    int checksum;
};

struct data_pkt{
    int sr_no;
    char msg[100];
    long checkSum;
};


struct ack_pkt{
    int ack;
    int sr_no;
    long checkSum;
};

void setZero(struct control_pkt * pkt){
    pkt->sr_no=0;
    pkt->no_of_pkt=0;
    pkt->checksum=0;
    pkt->end_sr_no=0;
    pkt->start_sr_no=0;
}

int noOfPck(char * msg){
    int len = strlen(msg);
    int req = len/100;
    double reqD = len/(double)100;
    if(reqD > req) req++;
    return req;
}

long createSumControl(struct control_pkt * c_pkt){
    long sum=0;
    sum+=c_pkt->end_sr_no;
    sum+=c_pkt->no_of_pkt;
    sum+=c_pkt->sr_no;
    sum+=c_pkt->start_sr_no;
    return sum;
}

int checkSumControl(struct control_pkt * c_pkt){
    long check = createSumControl(c_pkt);
    if(c_pkt->checksum == check) return 1;
    return 0;
}

long createSumACK(struct ack_pkt* ack){
    return ack->ack+ack->sr_no;
}

int checkSumACK(struct ack_pkt * ack){
    long ackSum = createSumACK(ack);
    if(ackSum == ack->checkSum) return 1;
    return 0;
}

long createSum(struct data_pkt* data){
    long msgSum=0;

    for(int i=0;i<100;i+=2){
          msgSum += ((data->msg[i]<<8) + data->msg[i+1]);
    }

    msgSum += data->sr_no;

    return msgSum;
}

int checkSum(struct data_pkt* data){
    long check = createSum(data);

    if(data->checkSum == check)return 1;
    return 0;
}
```

```
int shakeHand(int socket_desc,struct sockaddr_in client_addr,int client_addr_size,
    struct control_pkt * c_pkt){

    setZero(c_pkt);

    struct ack_pkt ackPkt;
    struct ack_pkt * ack = &ackPkt;

    ack->ack=0;
    ack->sr_no=0;

    int valid = 0;

    while(valid == 0){
            if(recvfrom(socket_desc,c_pkt,sizeof(*c_pkt),0,
                    (struct sockaddr *)&client_addr,&client_addr_size)<0){
                            printf("\nRecv error.\n");
                            return -1;
            }
            printf("\nRecved control pkt\n");
            valid = checkSumControl(c_pkt);
            ack->ack=1;
            ack->checkSum = createSumACK(ack);
            if(sendto(socket_desc,ack,sizeof(*ack),0,
                    (struct sockaddr *)&client_addr,client_addr_size)<0){
                            printf("\nError while sending akc to client.");
                            return -1;
            }
    }

    printf("\nSent control pkt ack.\n");
    return 1;

}

int comunicate(int socket_desc,struct sockaddr_in client_addr,int client_addr_size){
    struct data_pkt data;
    struct data_pkt * data1 = &data;
    memset(data1->msg,0,100);


    struct ack_pkt ackPKT;
    struct ack_pkt * ack = &ackPKT;


    if(recvfrom(socket_desc,data1,sizeof(*data1),0,
        (struct sockaddr *)&client_addr,&client_addr_size)<0){
                printf("\nRecv error.\n");
                return -1;
    }
    int valid = checkSum(data1);
    printf("\nMsg from client: %s",data1->msg);

    if(valid == 1){
        ack->ack=1;
        printf("\nData pkt is valid\n");
    }
    else{
        ack->ack=0;
        printf("\nData pkt is not valid\n");
    }
    ack->checkSum = createSumACK(ack);
    char c;
    scanf("%c",&c);

    if(sendto(socket_desc,ack,sizeof(*ack),0,
        (struct sockaddr *)&client_addr,client_addr_size)<0){
                printf("\nError while sending ack to client.");
                return -1;
    }


    if(sendto(socket_desc,ack,sizeof(*ack),0,
        (struct sockaddr *)&client_addr,client_addr_size)<0){
                printf("\nError while sending ack to client.");
                return -1;
    }

}

int comunicatePkt(int socket_desc,struct sockaddr_in client_addr,int client_addr_size,int
sr_no){
    struct data_pkt data;
    struct data_pkt * data1 = &data;
    memset(data1->msg,0,100);

    struct ack_pkt ackPKT;
    struct ack_pkt * ack = &ackPKT;

    if(recvfrom(socket_desc,data1,sizeof(*data1),0,
        (struct sockaddr *)&client_addr,&client_addr_size)<0){
                printf("\nRecv error.\n");
```

```
                    return -1;
    }
    int valid = checkSum(data1);
    printf("\nchecksum: %ld",createSum(data1));
    printf("\nMsg from client: %s",data1->msg);

    if(valid == 1){
            if(data1->sr_no != sr_no) return -1;
            ack->ack=1;
            printf("\nData pkt is valid\n");
    }
    else{
            ack->ack=-1;
            printf("\nData pkt is not valid\n");
    }
    ack->checkSum = createSumACK(ack);
    char c;
    scanf("%c",&c);


    if(sendto(socket_desc,ack,sizeof(*ack),0,
            (struct sockaddr *)&client_addr,client_addr_size)<0){
                    printf("\nError while sending ack to client.");
                    return -1;
    }
    return ack->ack;
}

int main(){
    int socket_desc;
    struct sockaddr_in server_addr,client_addr;
    int client_addr_size = sizeof(client_addr);

    socket_desc = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);
    if(socket_desc < 0){
            printf("\nError while creating socket.");
            return -1;
    }
    printf("\nCreated socket.");


    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(PORT);
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    //binding to port and ip
    if(bind(socket_desc,(struct sockaddr *)&server_addr,sizeof(server_addr))<0){
            printf("\nBinding error");
            return -1;
    }
    printf("\nBinding of socket.");

    printf("\nNow listening...");

    struct control_pkt pkt;


    int handshakeResult = shakeHand(socket_desc,client_addr,client_addr_size,&pkt);
    while(handshakeResult==-1){
            return -1;
    }

    printf("\nPackets : %d\n",pkt.no_of_pkt);

    for(int i=0;i<pkt.no_of_pkt;i++){
            int result = comunicatePkt(socket_desc,client_addr,client_addr_size,i);
            while(result == -1){
                    result = comunicatePkt(socket_desc,client_addr,client_addr_size,i);
            }
    }
    return 1;
}
```

Server Screenshot :



```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q1$ gcc server.c
saif@saif:~/Desktop/Computer Networks/Lab 03/Q1$ ./a.out

Created socket.
Binding of socket.
Now listening...
Recved control pkt

Sent control pkt ack.

Packets : 2

checksum: 82897
Msg from client: hello
Data pkt is valid
```

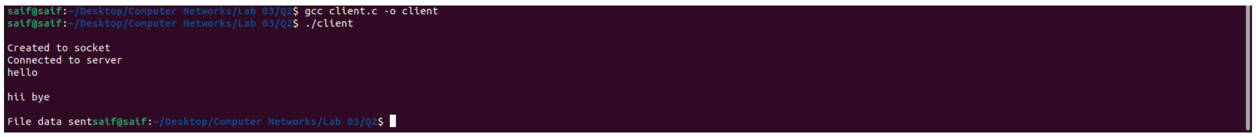
Q2 : TCP

Client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#define SIZE 1024
#define IP "127.0.0.1"
#define PORT 5050

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

	while(fgets(data,SIZE,fp) != NULL){
	printf("\n%s",data);
	if(send(sockfd,data,sizeof(data),0)==-1){
	printf("\nError while sending file.");
	return;
	}
	memset(data,0,SIZE);
	}

}

int main(){
	int sockfd,clientfd;
	struct sockaddr_in server_addr;
	FILE *fp;
	char *filename = "test.txt";

	sockfd = socket(AF_INET,SOCK_STREAM,0);
	if(sockfd < 0){
	printf("\nError while creating socket");
	return -1;
	}
	printf("\nCreated to socket");

	server_addr.sin_family = AF_INET;
	server_addr.sin_port = htons(PORT);
	server_addr.sin_addr.s_addr = inet_addr(IP);
```

```
        clientfd = connect(sockfd,(struct sockaddr *)&server_addr,sizeof(server_addr));
        if(clientfd == -1){
        printf("\nError while connecting");
        return -1;
        }
        printf("\nConnected to server");

        fp=fopen(filename,"r");
        if(fp==NULL){
        printf("\nError while reading file");
        return -1;
        }

        send_file(fp,sockfd);
        printf("\nFile data sent");

        return 0;

}
```

Client Screenshot :

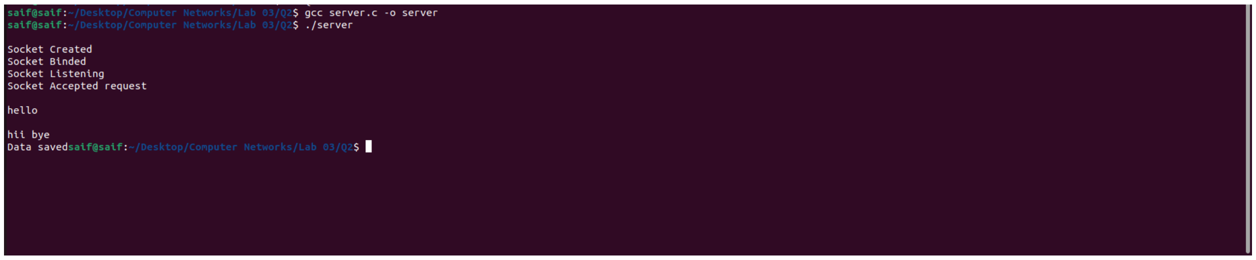


```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ gcc client.c -o client
saif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ ./client

Created to socket
Connected to server
hello

hii bye

File data sentsaif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ 
```

Server.c

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

#define PORT 5050
#define SIZE 1024

void write_file(int sockfd){
        int n;
        FILE *fp;
        char * filename = "get.txt";
        char buffer[SIZE];

        fp = fopen(filename,"w");
        while(1){
        n = recv(sockfd,buffer,SIZE,0);
        if(n<=0){break;return;}
        printf("\n%s",buffer);
        fprintf(fp,"%s",buffer);
        memset(buffer,0,1024);

        }
        return;
}

int main(){
    int server_fd,new_socket,valRead;
    struct sockaddr_in address;
    int opt = 1;
    int addrlen = sizeof(address);
    char buffer[1024];
    char *hello = "Hello from server";

    if((server_fd = socket(AF_INET,SOCK_STREAM,0))==0){
          printf("\nSocket creation error");
          return -1;
    }

    printf("\nSocket Created");
    if(setsockopt(server_fd,SOL_SOCKET,SO_REUSEADDR |
SO_REUSEPORT,&opt,sizeof(opt))){
        printf("\nSetsockopt");
        return -1;
    }

    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(PORT);

    if(bind(server_fd,(struct sockaddr *)&address , sizeof(address))<0){
        printf("\nBinding Error");
        return -1;
    }
    printf("\nSocket Binded");

    if(listen(server_fd,3)<0){
        nnections
        printf("\nListening Error");
        return -1;
    }
    printf("\nSocket Listening");

    if((new_socket = accept(server_fd,(struct sockaddr *)&address, (socklen_t
*)&addrlen))<0){
        epting clients connection
        printf("\nAccepting Error");
        return -1;
}
    printf("\nSocket Accepted request\n");
    write_file(new_socket);
        printf("Data saved");

    return 0;
}
```

Screenshot :



```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ gcc server.c -o server
saif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ ./server

Socket Created
Socket Binded
Socket Listening
Socket Accepted request

hello

hii bye
Data savedsaif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ 
```

Q3 : UDP

Client.c

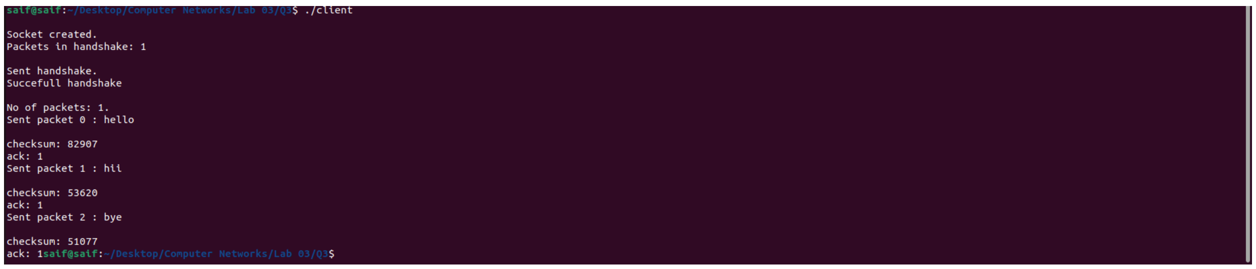
```
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdio.h>

#define PORT 5100


struct control_pkt{
    int sr_no;
    int no_of_pkt;
    int start_sr_no;
    int end_sr_no;
    int checksum;
};


struct data_pkt{
    int sr_no;
    char msg[100];
    long checkSum;
};

struct ack_pkt{
    int ack;
    int sr_no;
    long checkSum;
};


int noOfPck(int size){
    int req = size/100;
    double reqD = size/(double)100;
    if(reqD > req) req++;
    return req;
}


long createSumControl(struct control_pkt * c_pkt){
    long sum=0;
    sum+=c_pkt->end_sr_no;
    sum+=c_pkt->no_of_pkt;
    sum+=c_pkt->sr_no;
    sum+=c_pkt->start_sr_no;
    return sum;
}

int checkSumControl(struct control_pkt * c_pkt){
    long check = createSumControl(c_pkt);
    if(c_pkt->checksum == check) return 1;
    return 0;
}

long createSumACK(struct ack_pkt* ack){
    return ack->ack+ack->sr_no;
}

int checkSumACK(struct ack_pkt * ack){
    long ackSum = createSumACK(ack);
    if(ackSum == ack->checkSum) return 1;
    return 0;
}

long createSum(struct data_pkt* data){
    long msgSum=0;

    for(int i=0;i<100;i+=2){
            msgSum += ((data->msg[i]<<8) + data->msg[i+1]);
    }

    msgSum += data->sr_no;

    return msgSum;
}

int checkSum(struct data_pkt* data){
    long check = createSum(data);

    if(data->checkSum == check)return 1;
    return 0;
}


int shakeHand(int sock_fd,struct sockaddr_in server_addr,int sock_length,int size,
    struct control_pkt* c_pkt){

    int pkts = noOfPck(size);
    printf("\nPackets in handshake: %d\n",pkts);

    c_pkt->end_sr_no = pkts;
    c_pkt->sr_no = 0;
    c_pkt->no_of_pkt = pkts;
    c_pkt->start_sr_no=1;
    c_pkt->checksum = createSumControl(c_pkt);

    struct ack_pkt ackPkt;
    struct ack_pkt * ack = &ackPkt;

    ack->ack=-1;


    while(ack->ack != 1){
        if(sendto(sock_fd,c_pkt,sizeof(*c_pkt),0,
                (struct sockaddr *)&server_addr,sock_length)<0){
                    printf("\nError while sending message to server.\n");
                    return -1;
        }
        printf("\nSent handshake.");

        //time delay check
        struct timeval t;
        t.tv_sec=5;
        fd_set socks;
        FD_ZERO(&socks);
        FD_SET(sock_fd,&socks);

        if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
                recvfrom(sock_fd,ack,sizeof(*ack),0,
                (struct sockaddr *)&server_addr,&sock_length)<0){
                    printf("\nError in delay recv function.");
        }

    }
    return 1;

}


int comunicate(int sock_fd,struct sockaddr_in server_addr,int sock_length){
    struct data_pkt dataSend;
    struct data_pkt * data1 = &dataSend;
    memset(data1->msg,0,100);
    printf("\nEnter msg: ");

    fgets(data1->msg,100,stdin);
    data1->sr_no=1;

    data1->checkSum = createSum(data1);


    if(sendto(sock_fd,data1,sizeof(*data1),0,
        (struct sockaddr *)&server_addr,sock_length)<0){
            printf("\nError while sending message to server.");
            return -1;
    }
    printf("\nSent: %s",data1->msg);


    struct ack_pkt dataRecv;
    struct ack_pkt * ack = &dataRecv;
    ack->ack=-1;
    struct timeval t;
    t.tv_sec=5;
    fd_set socks;
    FD_ZERO(&socks);
    FD_SET(sock_fd,&socks);

    if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
        recvfrom(sock_fd,ack,sizeof(*ack),0,
        (struct sockaddr *)&server_addr,&sock_length)<0){
            printf("\nError in delay recv function.");

    }

    if(ack->ack==-1){
        printf("\nNo ack recved");
    }
    else{
        printf("\nAck recved");
    }
;

    return 1;
}
```

```
int comunicatePkt(int sock_fd,struct sockaddr_in server_addr,int sock_length,char *
msg,int sr_no){
    struct data_pkt dataSend;
    struct data_pkt * data = &dataSend;
    memset(data->msg,0,100);

    struct ack_pkt ackPkt;
    struct ack_pkt * ack = &ackPkt;

    ack->ack=-1;

    int i=0;
    while(*msg != '\0'){
            data->msg[i]=*msg;
            msg++;i++;
    }

    data->sr_no = sr_no;
    data->checkSum = createSum(data);

    if(sendto(sock_fd,data,sizeof(*data),0,
            (struct sockaddr *)&server_addr,sock_length)<0){
                    printf("\nError while sending message to server");
                    return -1;
    }

    printf("\nSent packet %d : %s ",sr_no,data->msg);
    printf("\nchecksum: %ld",data->checkSum);

    struct timeval t;
    t.tv_sec=5;
    fd_set socks;
    FD_ZERO(&socks);
    FD_SET(sock_fd,&socks);
    if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
            recvfrom(sock_fd,ack,sizeof(*ack),0,
            (struct sockaddr *)&server_addr,&sock_length)<0){
                    printf("\nError in delay recv function.");
    }
    printf("\nack: %d",ack->ack);
    return ack->ack;

}


int main(){
```

```
int sock_fd;
struct sockaddr_in server_addr;
int sock_length = sizeof(server_addr);

sock_fd = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);

if(sock_fd < 0){
        printf("\nError while creating socket.");
        return -1;
}
printf("\nSocket created.");

server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
server_addr.sin_port = htons(PORT);
server_addr.sin_family = AF_INET;

char msg[100];
int size = sizeof(msg)/sizeof(msg[0]);
memset(msg,0,100);

struct control_pkt pkt;

int handshakeResult = shakeHand(sock_fd,server_addr,sock_length,size,&pkt);
while(handshakeResult==-1){
        handshakeResult=shakeHand(sock_fd,server_addr,sock_length,size,&pkt);
}

printf("\nSuccefull handshake\n");
printf("\nNo of packets: %d.",pkt.no_of_pkt);


char * filename="test.txt";
FILE *fp = fopen(filename,"r");
if(fp == NULL) {
        printf("\nError while reading file");
}



int i=0;
while(fgets(msg,100,fp) !=NULL){
        int result = comunicatePkt(sock_fd,server_addr,sock_length,msg,i);
        while(result == -1){
                result = comunicatePkt(sock_fd,server_addr,sock_length,msg,i);
        }
        i++;
```

```
    }

    return 0;
}
```

Client Screenshot



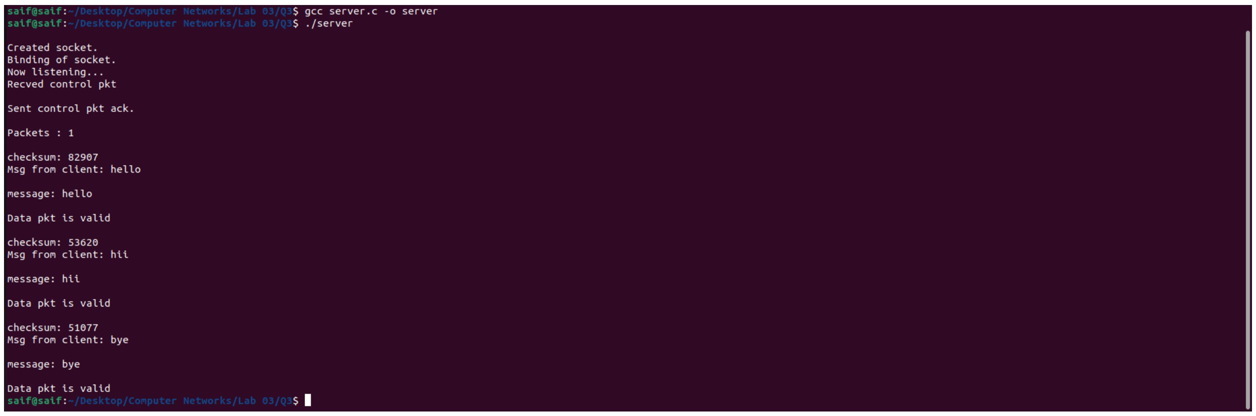
```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$ ./client

Socket created.
Packets in handshake: 1

Sent handshake.
Succefull handshake

No of packets: 1.
Sent packet 0 : hello

checksum: 82907
ack: 1
Sent packet 1 : hii

checksum: 53620
ack: 1
Sent packet 2 : bye

checksum: 51077
ack: 1saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$
```

Server.c

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

#define PORT 5100

struct control_pkt{
    int sr_no;
    int no_of_pkt;
    int start_sr_no;
    int end_sr_no;
    int checksum;
};

struct data_pkt{
    int sr_no;
    char msg[100];
    long checkSum;
};


struct ack_pkt{
    int ack;
    int sr_no;
    long checkSum;
};

void setZero(struct control_pkt * pkt){
    pkt->sr_no=0;
    pkt->no_of_pkt=0;
    pkt->checksum=0;
    pkt->end_sr_no=0;
    pkt->start_sr_no=0;
}

int noOfPck(char * msg){
    int len = strlen(msg);
    int req = len/100;
    double reqD = len/(double)100;
    if(reqD > req) req++;
    return req;
}
```

```
long createSumControl(struct control_pkt * c_pkt){
    long sum=0;
    sum+=c_pkt->end_sr_no;
    sum+=c_pkt->no_of_pkt;
    sum+=c_pkt->sr_no;
    sum+=c_pkt->start_sr_no;
    return sum;
}

int checkSumControl(struct control_pkt * c_pkt){
    long check = createSumControl(c_pkt);
    if(c_pkt->checksum == check) return 1;
    return 0;
}

long createSumACK(struct ack_pkt* ack){
    return ack->ack+ack->sr_no;
}

int checkSumACK(struct ack_pkt * ack){
    long ackSum = createSumACK(ack);
    if(ackSum == ack->checkSum) return 1;
    return 0;
}

long createSum(struct data_pkt* data){
    long msgSum=0;

    for(int i=0;i<100;i+=2){
            msgSum += ((data->msg[i]<<8) + data->msg[i+1]);
    }

    msgSum += data->sr_no;

    return msgSum;
}

int checkSum(struct data_pkt* data){
    long check = createSum(data);

    if(data->checkSum == check)return 1;
    return 0;
}

int shakeHand(int socket_desc,struct sockaddr_in client_addr,int client_addr_size,
}


    struct control_pkt * c_pkt){
    setZero(c_pkt);

    struct ack_pkt ackPkt;
    struct ack_pkt * ack = &ackPkt;

    ack->ack=0;
    ack->sr_no=0;

    int valid = 0;

    while(valid == 0){
            if(recvfrom(socket_desc,c_pkt,sizeof(*c_pkt),0,
                    (struct sockaddr *)&client_addr,&client_addr_size)<0){
                            printf("\nRecv error.\n");
                            return -1;
            }
            printf("\nRecved control pkt\n");
            valid = checkSumControl(c_pkt);
            ack->ack=1;
            ack->checkSum = createSumACK(ack);
            if(sendto(socket_desc,ack,sizeof(*ack),0,
                    (struct sockaddr *)&client_addr,client_addr_size)<0){
                            printf("\nError while sending akc to client.");
                            return -1;
            }
    }

    printf("\nSent control pkt ack.\n");
    return 1;

}

int comunicate(int socket_desc,struct sockaddr_in client_addr,int client_addr_size){
    struct data_pkt data;
    struct data_pkt * data1 = &data;
    memset(data1->msg,0,100);

    struct ack_pkt ackPKT;
    struct ack_pkt * ack = &ackPKT;

    if(recvfrom(socket_desc,data1,sizeof(*data1),0,
            (struct sockaddr *)&client_addr,&client_addr_size)<0){
                    printf("\nRecv error.\n");
                    return -1;
    }
    int valid = checkSum(data1);
    printf("\nMsg from client: %s",data1->msg);

    if(valid == 1){
        ack->ack=1;
        printf("\nData pkt is valid\n");
    }
    else{
        ack->ack=0;
        printf("\nData pkt is not valid\n");
    }
    ack->checkSum = createSumACK(ack);
    char c;
    scanf("%c",&c);

    if(sendto(socket_desc,ack,sizeof(*ack),0,
        (struct sockaddr *)&client_addr,client_addr_size)<0){
            printf("\nError while sending ack to client.");
            return -1;
    }


    if(sendto(socket_desc,ack,sizeof(*ack),0,
        (struct sockaddr *)&client_addr,client_addr_size)<0){
            printf("\nError while sending ack to client.");
            return -1;
    }

}

int comunicatePkt(int socket_desc,struct sockaddr_in client_addr,int client_addr_size,int
sr_no){
    struct data_pkt data;
    struct data_pkt * data1 = &data;
    memset(data1->msg,0,100);

    struct ack_pkt ackPKT;
    struct ack_pkt * ack = &ackPKT;

    if(recvfrom(socket_desc,data1,sizeof(*data1),0,
        (struct sockaddr *)&client_addr,&client_addr_size)<0){
            printf("\nRecv error.\n");
            return -1;
    }
    int valid = checkSum(data1);
    printf("\nchecksum: %ld",createSum(data1));
    printf("\nMsg from client: %s",data1->msg);

    if(valid == 1){
        if(data1->sr_no != sr_no) return -1;
        ack->ack=1;

        char * filename = "get.txt";
        FILE *fp = fopen(filename,"a");

        fprintf(fp,"%s",data1->msg);
        printf("\nmessage: %s",data1->msg);

        printf("\nData pkt is valid\n");
    }
    else{
        ack->ack=-1;
        printf("\nData pkt is not valid\n");
    }
    ack->checkSum = createSumACK(ack);

    if(sendto(socket_desc,ack,sizeof(*ack),0,
        (struct sockaddr *)&client_addr,client_addr_size)<0){
            printf("\nError while sending ack to client.");
            return -1;
    }
    return ack->ack;
}

int main(){
    int socket_desc;
    struct sockaddr_in server_addr,client_addr;
    int client_addr_size = sizeof(client_addr);

    socket_desc = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);
    if(socket_desc < 0){
        printf("\nError while creating socket.");
        return -1;
    }
    printf("\nCreated socket.");

    p
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(PORT);
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");

    if(bind(socket_desc,(struct sockaddr *)&server_addr,sizeof(server_addr))<0){
        printf("\nBinding error");
        return -1;
    }
    printf("\nBinding of socket.");

    printf("\nNow listening...");

    struct control_pkt pkt;


    int handshakeResult = shakeHand(socket_desc,client_addr,client_addr_size,&pkt);
    while(handshakeResult==-1){
        handshakeResult=shakeHand(socket_desc,client_addr,client_addr_size,&pkt);
        return -1;
    }

    printf("\nPackets : %d\n",pkt.no_of_pkt);

    for(int i=0;i<3;i++){
        int result = comunicatePkt(socket_desc,client_addr,client_addr_size,i);
        while(result == -1){
                result = comunicatePkt(socket_desc,client_addr,client_addr_size,i);
        }
    }
    return 1;
}
```

Server Screenshot :



```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$ gcc server.c -o server
saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$ ./server

Created socket.
Binding of socket.
Now listening...
Recved control pkt

Sent control pkt ack.

Packets : 1

checksum: 82907
Msg from client: hello

message: hello

Data pkt is valid

checksum: 53620
Msg from client: hii

message: hii

Data pkt is valid

checksum: 51077
Msg from client: bye

message: bye

Data pkt is valid
saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$ 
```

202212083

1: Time Control

Client.c

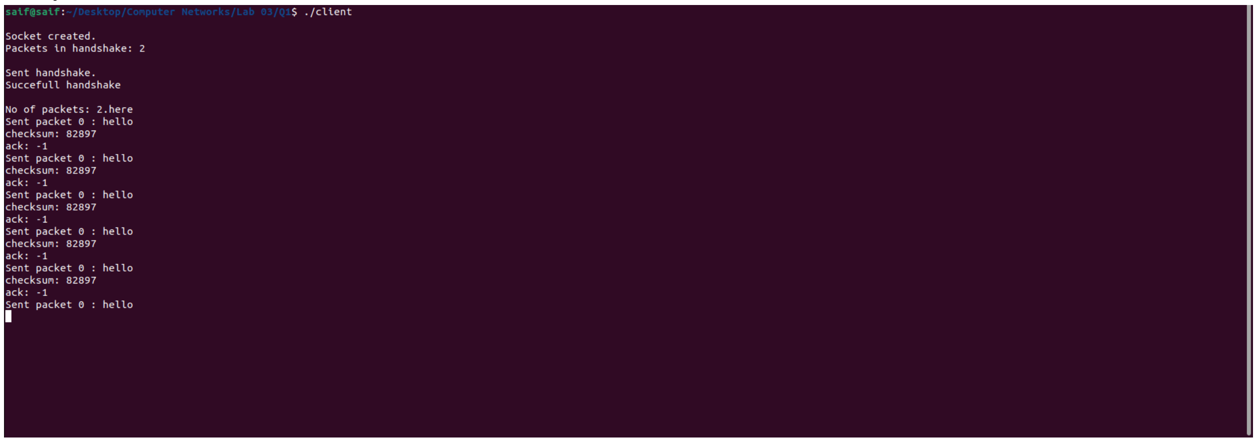
```
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdio.h>

#define PORT 5100


struct control_pkt{
    int sr_no;
    int no_of_pkt;
    int start_sr_no;
    int end_sr_no;
    int checksum;
};


struct data_pkt{
    int sr_no;
    char msg[100];
    long checkSum;
};

struct ack_pkt{
    int ack;
    int sr_no;
    long checkSum;
};


int noOfPck(int size){
    int req = size/100;
    double reqD = size/(double)100;
    if(reqD > req) req++;
    return req;
}


long createSumControl(struct control_pkt * c_pkt){
    long sum=0;
    sum+=c_pkt->end_sr_no;
    sum+=c_pkt->no_of_pkt;
    sum+=c_pkt->sr_no;
    sum+=c_pkt->start_sr_no;
    return sum;
}

int checkSumControl(struct control_pkt * c_pkt){
    long check = createSumControl(c_pkt);
    if(c_pkt->checksum == check) return 1;
    return 0;
}

long createSumACK(struct ack_pkt* ack){
    return ack->ack+ack->sr_no;
}

int checkSumACK(struct ack_pkt * ack){
    long ackSum = createSumACK(ack);
    if(ackSum == ack->checkSum) return 1;
    return 0;
}

long createSum(struct data_pkt* data){
    long msgSum=0;

    for(int i=0;i<100;i+=2){
        msgSum += ((data->msg[i]<<8) + data->msg[i+1]);
    }

    msgSum += data->sr_no;

    return msgSum;
}

int checkSum(struct data_pkt* data){
    long check = createSum(data);

    if(data->checkSum == check)return 1;
    return 0;
}


int shakeHand(int sock_fd,struct sockaddr_in server_addr,int sock_length,int size,
    struct control_pkt* c_pkt){

    int pkts = noOfPck(size);

    printf("\nPackets in handshake: %d\n",pkts);

    c_pkt->end_sr_no = pkts;
    c_pkt->sr_no = 0;
    c_pkt->no_of_pkt = pkts;
    c_pkt->start_sr_no=1;
    c_pkt->checksum = createSumControl(c_pkt);

    //ack
    struct ack_pkt ackPkt;
    struct ack_pkt * ack = &ackPkt;

    ack->ack=-1;


    while(ack->ack != 1){
        if(sendto(sock_fd,c_pkt,sizeof(*c_pkt),0,
                (struct sockaddr *)&server_addr,sock_length)<0){
                    printf("\nError while sending message to server.\n");
                    return -1;
        }
        printf("\nSent handshake.");

        struct timeval t;
        t.tv_sec=5;
        fd_set socks;
        FD_ZERO(&socks);
        FD_SET(sock_fd,&socks);

        if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
                recvfrom(sock_fd,ack,sizeof(*ack),0,
                (struct sockaddr *)&server_addr,&sock_length)<0){
                    printf("\nError in delay recv function.");
        }

    }
    return 1;
```

```
}


int comunicate(int sock_fd,struct sockaddr_in server_addr,int sock_length){
   //msg
   struct data_pkt dataSend;
   struct data_pkt * data1 = &dataSend;
   memset(data1->msg,0,100);

   printf("\nEnter msg: ");

   fgets(data1->msg,100,stdin);
   data1->sr_no=1;

   data1->checkSum = createSum(data1);


   if(sendto(sock_fd,data1,sizeof(*data1),0,
         (struct sockaddr *)&server_addr,sock_length)<0){
                 printf("\nError while sending message to server.");
                 return -1;
   }
   printf("\nSent: %s",data1->msg);


   struct ack_pkt dataRecv;
   struct ack_pkt * ack = &dataRecv;
   ack->ack=-1;

   struct timeval t;
   t.tv_sec=5;
   fd_set socks;
   FD_ZERO(&socks);
   FD_SET(sock_fd,&socks);

   if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
         recvfrom(sock_fd,ack,sizeof(*ack),0,
         (struct sockaddr *)&server_addr,&sock_length)<0){
                 printf("\nError in delay recv function.");

   }

   if(ack->ack==-1){
         printf("\nNo ack recved");
   }
```

```
    else{
          printf("\nAck recved");
    }


    return 1;
}

int comunicatePkt(int sock_fd,struct sockaddr_in server_addr,int sock_length,char *
msg,int sr_no){

    struct data_pkt dataSend;
    struct data_pkt * data = &dataSend;
    memset(data->msg,0,100);


    struct ack_pkt ackPkt;
    struct ack_pkt * ack = &ackPkt;

    ack->ack=-1;

    int i=0;
    while(*msg != '\0'){
          data->msg[i]=*msg;
          msg++;i++;
    }

    data->sr_no = sr_no;
    data->checkSum = createSum(data);

    if(sendto(sock_fd,data,sizeof(*data),0,
          (struct sockaddr *)&server_addr,sock_length)<0){
                printf("\nError while sending message to server");
                return -1;
    }

    printf("\nSent packet %d : %s ",sr_no,data->msg);
    printf("\nchecksum: %ld",data->checkSum);

    struct timeval t;
    t.tv_sec=5;
    fd_set socks;
    FD_ZERO(&socks);
    FD_SET(sock_fd,&socks);
    if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
          recvfrom(sock_fd,ack,sizeof(*ack),0,
        (struct sockaddr *)&server_addr,&sock_length)<0){
                printf("\nError in delay recv function.");
    }
    printf("\nack: %d",ack->ack);
    return ack->ack;

}


int main(){
    int sock_fd;
    struct sockaddr_in server_addr;
    int sock_length = sizeof(server_addr);

    sock_fd = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);

    if(sock_fd < 0){
        printf("\nError while creating socket.");
        return -1;
    }
    printf("\nSocket created.");

    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    server_addr.sin_port = htons(PORT);
    server_addr.sin_family = AF_INET;

    char msg[200];
    int size = sizeof(msg)/sizeof(msg[0]);

    struct control_pkt pkt;

    int handshakeResult = shakeHand(sock_fd,server_addr,sock_length,size,&pkt);
    while(handshakeResult==-1){
        handshakeResult=shakeHand(sock_fd,server_addr,sock_length,size,&pkt);
    }

    printf("\nSuccefull handshake\n");
    printf("\nNo of packets: %d.",pkt.no_of_pkt);

    char *pkt_datas[2];
    char * f = "hello";
    char * s = "world";
    printf("here");
    pkt_datas[0]=f;
    pkt_datas[1]=s;
```

```
  for(int i=0;i<pkt.no_of_pkt;i++){
            int result = comunicatePkt(sock_fd,server_addr,sock_length,pkt_datas[i],i);
            while(result == -1){
                      result = comunicatePkt(sock_fd,server_addr,sock_length,pkt_datas[i],i);
            }
    }

    return 0;
}
```

Output:



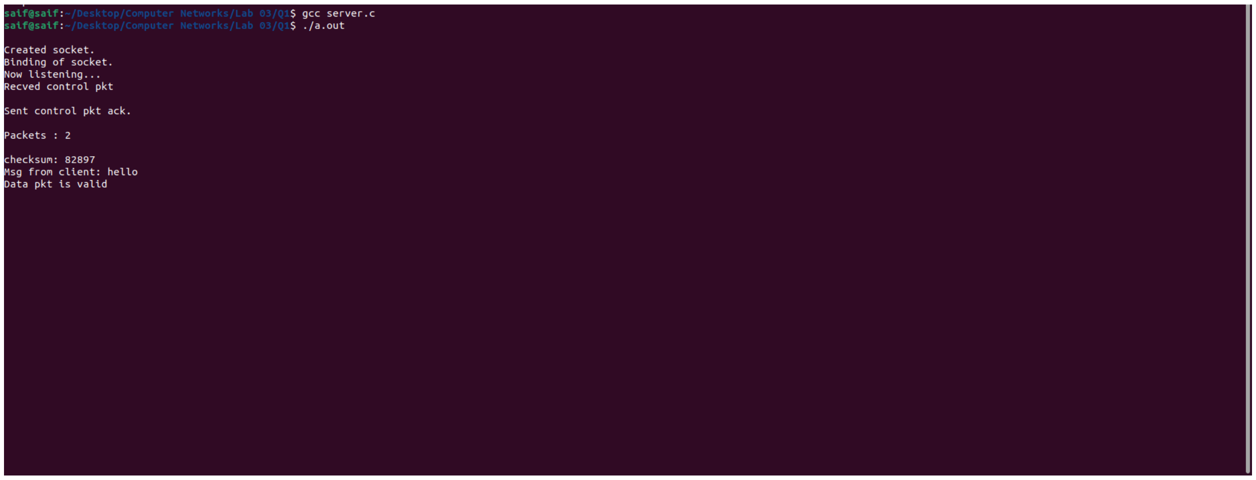
```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q1$ ./client

Socket created.
Packets in handshake: 2

Sent handshake.
Succefull handshake

No of packets: 2.here
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
checksum: 82897
ack: -1
Sent packet 0 : hello
```

Server.c

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

#define PORT 5100

struct control_pkt{
    int sr_no;
    int no_of_pkt;
    int start_sr_no;
    int end_sr_no;
    int checksum;
};

struct data_pkt{
    int sr_no;
    char msg[100];
    long checkSum;
};


struct ack_pkt{
    int ack;
    int sr_no;
    long checkSum;
};

void setZero(struct control_pkt * pkt){
    pkt->sr_no=0;
    pkt->no_of_pkt=0;
    pkt->checksum=0;
    pkt->end_sr_no=0;
    pkt->start_sr_no=0;
}

int noOfPck(char * msg){
    int len = strlen(msg);
    int req = len/100;
    double reqD = len/(double)100;
    if(reqD > req) req++;
    return req;
}

long createSumControl(struct control_pkt * c_pkt){
    long sum=0;
    sum+=c_pkt->end_sr_no;
    sum+=c_pkt->no_of_pkt;
    sum+=c_pkt->sr_no;
    sum+=c_pkt->start_sr_no;
    return sum;
}

int checkSumControl(struct control_pkt * c_pkt){
    long check = createSumControl(c_pkt);
    if(c_pkt->checksum == check) return 1;
    return 0;
}

long createSumACK(struct ack_pkt* ack){
    return ack->ack+ack->sr_no;
}

int checkSumACK(struct ack_pkt * ack){
    long ackSum = createSumACK(ack);
    if(ackSum == ack->checkSum) return 1;
    return 0;
}

long createSum(struct data_pkt* data){
    long msgSum=0;

    for(int i=0;i<100;i+=2){
            msgSum += ((data->msg[i]<<8) + data->msg[i+1]);
    }

    msgSum += data->sr_no;

    return msgSum;
}

int checkSum(struct data_pkt* data){
    long check = createSum(data);

    if(data->checkSum == check)return 1;
    return 0;
}
```

```
int shakeHand(int socket_desc,struct sockaddr_in client_addr,int client_addr_size,
   struct control_pkt * c_pkt){

   setZero(c_pkt);

   struct ack_pkt ackPkt;
   struct ack_pkt * ack = &ackPkt;

   ack->ack=0;
   ack->sr_no=0;

   int valid = 0;

   while(valid == 0){
         if(recvfrom(socket_desc,c_pkt,sizeof(*c_pkt),0,
                  (struct sockaddr *)&client_addr,&client_addr_size)<0){
                        printf("\nRecv error.\n");
                        return -1;
         }
         printf("\nRecved control pkt\n");
         valid = checkSumControl(c_pkt);
         ack->ack=1;
         ack->checkSum = createSumACK(ack);
         if(sendto(socket_desc,ack,sizeof(*ack),0,
                  (struct sockaddr *)&client_addr,client_addr_size)<0){
                        printf("\nError while sending akc to client.");
                        return -1;
         }
   }

   printf("\nSent control pkt ack.\n");
   return 1;

}

int comunicate(int socket_desc,struct sockaddr_in client_addr,int client_addr_size){
   struct data_pkt data;
   struct data_pkt * data1 = &data;
   memset(data1->msg,0,100);


   struct ack_pkt ackPKT;
   struct ack_pkt * ack = &ackPKT;


   if(recvfrom(socket_desc,data1,sizeof(*data1),0,
        (struct sockaddr *)&client_addr,&client_addr_size)<0){
                printf("\nRecv error.\n");
                return -1;
    }
    int valid = checkSum(data1);
    printf("\nMsg from client: %s",data1->msg);

    if(valid == 1){
        ack->ack=1;
        printf("\nData pkt is valid\n");
    }
    else{
        ack->ack=0;
        printf("\nData pkt is not valid\n");
    }
    ack->checkSum = createSumACK(ack);
    char c;
    scanf("%c",&c);

    if(sendto(socket_desc,ack,sizeof(*ack),0,
        (struct sockaddr *)&client_addr,client_addr_size)<0){
                printf("\nError while sending ack to client.");
                return -1;
    }


    if(sendto(socket_desc,ack,sizeof(*ack),0,
        (struct sockaddr *)&client_addr,client_addr_size)<0){
                printf("\nError while sending ack to client.");
                return -1;
    }

}

int comunicatePkt(int socket_desc,struct sockaddr_in client_addr,int client_addr_size,int
sr_no){
    struct data_pkt data;
    struct data_pkt * data1 = &data;
    memset(data1->msg,0,100);

    struct ack_pkt ackPKT;
    struct ack_pkt * ack = &ackPKT;

    if(recvfrom(socket_desc,data1,sizeof(*data1),0,
        (struct sockaddr *)&client_addr,&client_addr_size)<0){
                printf("\nRecv error.\n");
                return -1;
    }
    int valid = checkSum(data1);
    printf("\nchecksum: %ld",createSum(data1));
    printf("\nMsg from client: %s",data1->msg);

    if(valid == 1){
        if(data1->sr_no != sr_no) return -1;
        ack->ack=1;
        printf("\nData pkt is valid\n");
    }
    else{
        ack->ack=-1;
        printf("\nData pkt is not valid\n");
    }
    ack->checkSum = createSumACK(ack);
    char c;
    scanf("%c",&c);


    if(sendto(socket_desc,ack,sizeof(*ack),0,
        (struct sockaddr *)&client_addr,client_addr_size)<0){
                printf("\nError while sending ack to client.");
                return -1;
    }
    return ack->ack;
}

int main(){
    int socket_desc;
    struct sockaddr_in server_addr,client_addr;
    int client_addr_size = sizeof(client_addr);

    socket_desc = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);
    if(socket_desc < 0){
        printf("\nError while creating socket.");
        return -1;
    }
    printf("\nCreated socket.");


    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(PORT);
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    //binding to port and ip
    if(bind(socket_desc,(struct sockaddr *)&server_addr,sizeof(server_addr))<0){
        printf("\nBinding error");
        return -1;
    }
    printf("\nBinding of socket.");

    printf("\nNow listening...");

    struct control_pkt pkt;


    int handshakeResult = shakeHand(socket_desc,client_addr,client_addr_size,&pkt);
    while(handshakeResult==-1){
        return -1;
    }

    printf("\nPackets : %d\n",pkt.no_of_pkt);

    for(int i=0;i<pkt.no_of_pkt;i++){
        int result = comunicatePkt(socket_desc,client_addr,client_addr_size,i);
        while(result == -1){
            result = comunicatePkt(socket_desc,client_addr,client_addr_size,i);
        }
    }
    return 1;
}
```

Server Screenshot :



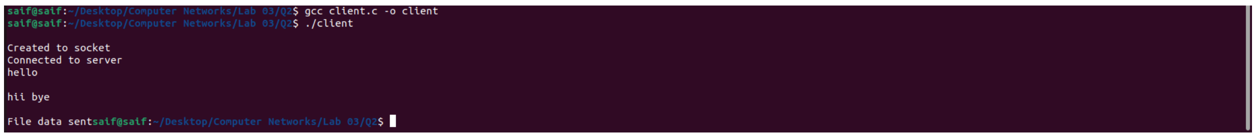
```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q1$ gcc server.c
saif@saif:~/Desktop/Computer Networks/Lab 03/Q1$ ./a.out

Created socket.
Binding of socket.
Now listening...
Recved control pkt

Sent control pkt ack.

Packets : 2

checksum: 82897
Msg from client: hello
Data pkt is valid
```

Q2 : TCP

Client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#define SIZE 1024
#define IP "127.0.0.1"
#define PORT 5050

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

        while(fgets(data,SIZE,fp) != NULL){
        printf("\n%s",data);
        if(send(sockfd,data,sizeof(data),0)==-1){
        printf("\nError while sending file.");
        return;
        }
        memset(data,0,SIZE);
        }

}

int main(){
        int sockfd,clientfd;
        struct sockaddr_in server_addr;
        FILE *fp;
        char *filename = "test.txt";

        sockfd = socket(AF_INET,SOCK_STREAM,0);
        if(sockfd < 0){
        printf("\nError while creating socket");
        return -1;
        }
        printf("\nCreated to socket");

        server_addr.sin_family = AF_INET;
        server_addr.sin_port = htons(PORT);
        server_addr.sin_addr.s_addr = inet_addr(IP);
        clientfd = connect(sockfd,(struct sockaddr *)&server_addr,sizeof(server_addr));
        if(clientfd == -1){
        printf("\nError while connecting");
        return -1;
        }
        printf("\nConnected to server");

        fp=fopen(filename,"r");
        if(fp==NULL){
        printf("\nError while reading file");
        return -1;
        }

        send_file(fp,sockfd);
        printf("\nFile data sent");

        return 0;

}
```

Client Screenshot :



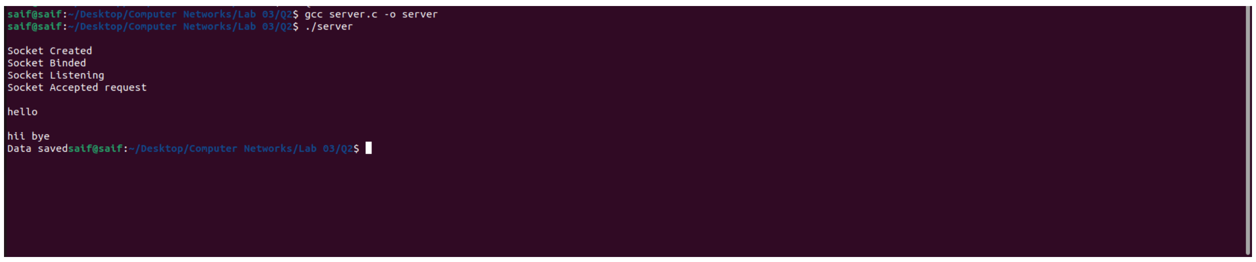
```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ gcc client.c -o client
saif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ ./client

Created to socket
Connected to server
hello

hii bye

File data sentsaif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ 
```

Server.c

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

#define PORT 5050
#define SIZE 1024

void write_file(int sockfd){
        int n;
        FILE *fp;
        char * filename = "get.txt";
        char buffer[SIZE];

        fp = fopen(filename,"w");
        while(1){
        n = recv(sockfd,buffer,SIZE,0);
        if(n<=0){break;return;}
        printf("\n%s",buffer);
        fprintf(fp,"%s",buffer);
        memset(buffer,0,1024);

        }
        return;
}

int main(){
    int server_fd,new_socket,valRead;
    struct sockaddr_in address;
    int opt = 1;
    int addrlen = sizeof(address);
    char buffer[1024];
    char *hello = "Hello from server";

    if((server_fd = socket(AF_INET,SOCK_STREAM,0))==0){
          printf("\nSocket creation error");
          return -1;
    }

    printf("\nSocket Created");
    if(setsockopt(server_fd,SOL_SOCKET,SO_REUSEADDR |
SO_REUSEPORT,&opt,sizeof(opt))){
```

```
            printf("\nSetsockopt");
            return -1;
    }

    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(PORT);

    if(bind(server_fd,(struct sockaddr *)&address , sizeof(address))<0){
            printf("\nBinding Error");
            return -1;
    }
    printf("\nSocket Binded");

    if(listen(server_fd,3)<0){
            nnections
            printf("\nListening Error");
            return -1;
    }
    printf("\nSocket Listening");

    if((new_socket = accept(server_fd,(struct sockaddr *)&address, (socklen_t
*)&addrlen))<0){
            epting clients connection
            printf("\nAccepting Error");
            return -1;
}
    printf("\nSocket Accepted request\n");
    write_file(new_socket);
           printf("Data saved");

    return 0;
}
```

Screenshot :



```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ gcc server.c -o server
saif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ ./server

Socket Created
Socket Binded
Socket Listening
Socket Accepted request

hello

hii bye
Data savedsaif@saif:~/Desktop/Computer Networks/Lab 03/Q2$ 
```

Q3 : UDP

Client.c

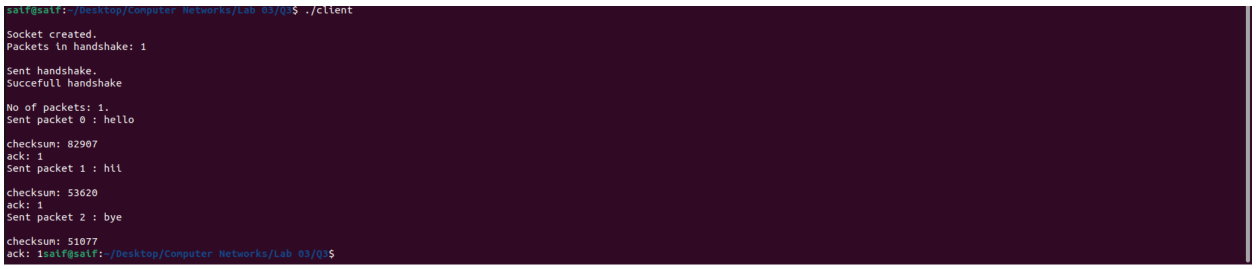
```
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdio.h>

#define PORT 5100


struct control_pkt{
    int sr_no;
    int no_of_pkt;
    int start_sr_no;
    int end_sr_no;
    int checksum;
};


struct data_pkt{
    int sr_no;
    char msg[100];
    long checkSum;
};

struct ack_pkt{
    int ack;
    int sr_no;
    long checkSum;
};


int noOfPck(int size){
    int req = size/100;
    double reqD = size/(double)100;
    if(reqD > req) req++;
    return req;
}


long createSumControl(struct control_pkt * c_pkt){
    long sum=0;
    sum+=c_pkt->end_sr_no;
    sum+=c_pkt->no_of_pkt;
    sum+=c_pkt->sr_no;
    sum+=c_pkt->start_sr_no;
    return sum;
}

int checkSumControl(struct control_pkt * c_pkt){
    long check = createSumControl(c_pkt);
    if(c_pkt->checksum == check) return 1;
    return 0;
}

long createSumACK(struct ack_pkt* ack){
    return ack->ack+ack->sr_no;
}

int checkSumACK(struct ack_pkt * ack){
    long ackSum = createSumACK(ack);
    if(ackSum == ack->checkSum) return 1;
    return 0;
}

long createSum(struct data_pkt* data){
    long msgSum=0;

    for(int i=0;i<100;i+=2){
        msgSum += ((data->msg[i]<<8) + data->msg[i+1]);
    }

    msgSum += data->sr_no;

    return msgSum;
}

int checkSum(struct data_pkt* data){
    long check = createSum(data);

    if(data->checkSum == check)return 1;
    return 0;
}


int shakeHand(int sock_fd,struct sockaddr_in server_addr,int sock_length,int size,
    struct control_pkt* c_pkt){

    int pkts = noOfPck(size);
    printf("\nPackets in handshake: %d\n",pkts);

    c_pkt->end_sr_no = pkts;
    c_pkt->sr_no = 0;
    c_pkt->no_of_pkt = pkts;
    c_pkt->start_sr_no=1;
    c_pkt->checksum = createSumControl(c_pkt);

    struct ack_pkt ackPkt;
    struct ack_pkt * ack = &ackPkt;

    ack->ack=-1;


    while(ack->ack != 1){
        if(sendto(sock_fd,c_pkt,sizeof(*c_pkt),0,
                (struct sockaddr *)&server_addr,sock_length)<0){
                    printf("\nError while sending message to server.\n");
                    return -1;
        }
        printf("\nSent handshake.");

        //time delay check
        struct timeval t;
        t.tv_sec=5;
        fd_set socks;
        FD_ZERO(&socks);
        FD_SET(sock_fd,&socks);

        if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
                recvfrom(sock_fd,ack,sizeof(*ack),0,
                (struct sockaddr *)&server_addr,&sock_length)<0){
                    printf("\nError in delay recv function.");
        }

    }
    return 1;

}


int comunicate(int sock_fd,struct sockaddr_in server_addr,int sock_length){
    struct data_pkt dataSend;
    struct data_pkt * data1 = &dataSend;
    memset(data1->msg,0,100);
    printf("\nEnter msg: ");

    fgets(data1->msg,100,stdin);
    data1->sr_no=1;

    data1->checkSum = createSum(data1);


    if(sendto(sock_fd,data1,sizeof(*data1),0,
            (struct sockaddr *)&server_addr,sock_length)<0){
                printf("\nError while sending message to server.");
                return -1;
    }
    printf("\nSent: %s",data1->msg);


    struct ack_pkt dataRecv;
    struct ack_pkt * ack = &dataRecv;
    ack->ack=-1;
    struct timeval t;
    t.tv_sec=5;
    fd_set socks;
    FD_ZERO(&socks);
    FD_SET(sock_fd,&socks);

    if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
            recvfrom(sock_fd,ack,sizeof(*ack),0,
            (struct sockaddr *)&server_addr,&sock_length)<0){
                printf("\nError in delay recv function.");

    }

    if(ack->ack==-1){
            printf("\nNo ack recved");
    }
    else{
            printf("\nAck recved");
    }
;

    return 1;
}
```

```
int comunicatePkt(int sock_fd,struct sockaddr_in server_addr,int sock_length,char *
msg,int sr_no){
    struct data_pkt dataSend;
    struct data_pkt * data = &dataSend;
    memset(data->msg,0,100);

    struct ack_pkt ackPkt;
    struct ack_pkt * ack = &ackPkt;

    ack->ack=-1;

    int i=0;
    while(*msg != '\0'){
            data->msg[i]=*msg;
            msg++;i++;
    }

    data->sr_no = sr_no;
    data->checkSum = createSum(data);

    if(sendto(sock_fd,data,sizeof(*data),0,
            (struct sockaddr *)&server_addr,sock_length)<0){
                    printf("\nError while sending message to server");
                    return -1;
    }

    printf("\nSent packet %d : %s ",sr_no,data->msg);
    printf("\nchecksum: %ld",data->checkSum);

    struct timeval t;
    t.tv_sec=5;
    fd_set socks;
    FD_ZERO(&socks);
    FD_SET(sock_fd,&socks);
    if(select(sock_fd+1,&socks,NULL,NULL,&t) &&
            recvfrom(sock_fd,ack,sizeof(*ack),0,
            (struct sockaddr *)&server_addr,&sock_length)<0){
                    printf("\nError in delay recv function.");
    }
    printf("\nack: %d",ack->ack);
    return ack->ack;

}


int main(){
```

```
int sock_fd;
struct sockaddr_in server_addr;
int sock_length = sizeof(server_addr);

sock_fd = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);

if(sock_fd < 0){
        printf("\nError while creating socket.");
        return -1;
}
printf("\nSocket created.");

server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
server_addr.sin_port = htons(PORT);
server_addr.sin_family = AF_INET;

char msg[100];
int size = sizeof(msg)/sizeof(msg[0]);
memset(msg,0,100);

struct control_pkt pkt;

int handshakeResult = shakeHand(sock_fd,server_addr,sock_length,size,&pkt);
while(handshakeResult==-1){
        handshakeResult=shakeHand(sock_fd,server_addr,sock_length,size,&pkt);
}

printf("\nSuccefull handshake\n");
printf("\nNo of packets: %d.",pkt.no_of_pkt);


char * filename="test.txt";
FILE *fp = fopen(filename,"r");
if(fp == NULL) {
        printf("\nError while reading file");
}



int i=0;
while(fgets(msg,100,fp) !=NULL){
        int result = comunicatePkt(sock_fd,server_addr,sock_length,msg,i);
        while(result == -1){
                result = comunicatePkt(sock_fd,server_addr,sock_length,msg,i);
        }
        i++;
```

```
    }

    return 0;
}
```

Client Screenshot



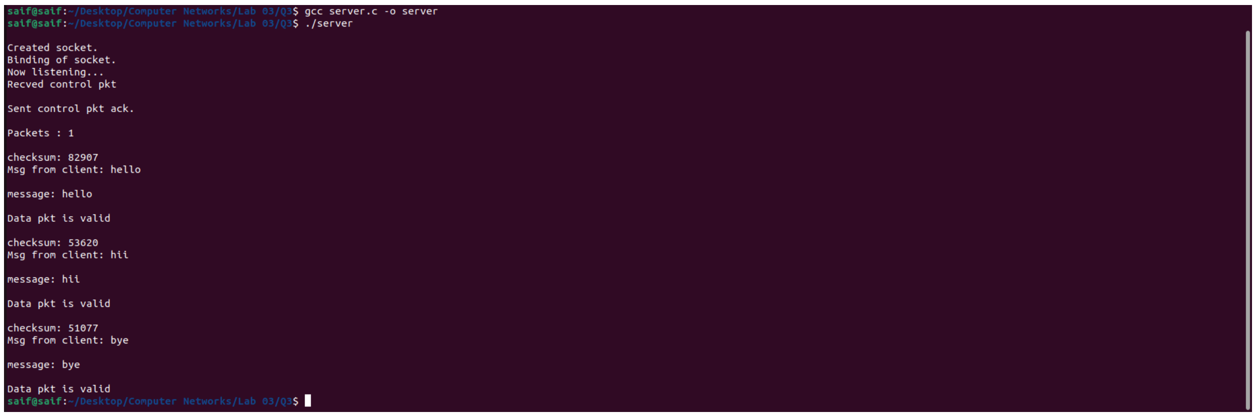
```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$ ./client

Socket created.
Packets in handshake: 1

Sent handshake.
Succefull handshake

No of packets: 1.
Sent packet 0 : hello

checksum: 82907
ack: 1
Sent packet 1 : hii

checksum: 53620
ack: 1
Sent packet 2 : bye

checksum: 51077
ack: 1saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$
```

Server.c

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

#define PORT 5100

struct control_pkt{
   int sr_no;
   int no_of_pkt;
   int start_sr_no;
   int end_sr_no;
   int checksum;
};

struct data_pkt{
   int sr_no;
   char msg[100];
   long checkSum;
};


struct ack_pkt{
   int ack;
   int sr_no;
   long checkSum;
};

void setZero(struct control_pkt * pkt){
   pkt->sr_no=0;
   pkt->no_of_pkt=0;
   pkt->checksum=0;
   pkt->end_sr_no=0;
   pkt->start_sr_no=0;
}

int noOfPck(char * msg){
   int len = strlen(msg);
   int req = len/100;
   double reqD = len/(double)100;
   if(reqD > req) req++;
   return req;
}
```

```
long createSumControl(struct control_pkt * c_pkt){
   long sum=0;
   sum+=c_pkt->end_sr_no;
   sum+=c_pkt->no_of_pkt;
   sum+=c_pkt->sr_no;
   sum+=c_pkt->start_sr_no;
   return sum;
}

int checkSumControl(struct control_pkt * c_pkt){
   long check = createSumControl(c_pkt);
   if(c_pkt->checksum == check) return 1;
   return 0;
}

long createSumACK(struct ack_pkt* ack){
   return ack->ack+ack->sr_no;
}

int checkSumACK(struct ack_pkt * ack){
   long ackSum = createSumACK(ack);
   if(ackSum == ack->checkSum) return 1;
   return 0;
}

long createSum(struct data_pkt* data){
   long msgSum=0;

   for(int i=0;i<100;i+=2){
         msgSum += ((data->msg[i]<<8) + data->msg[i+1]);
   }

   msgSum += data->sr_no;

   return msgSum;
}

int checkSum(struct data_pkt* data){
   long check = createSum(data);

   if(data->checkSum == check)return 1;
   return 0;
}

int shakeHand(int socket_desc,struct sockaddr_in client_addr,int client_addr_size,
   }


   struct control_pkt * c_pkt){
   setZero(c_pkt);

   struct ack_pkt ackPkt;
   struct ack_pkt * ack = &ackPkt;

   ack->ack=0;
   ack->sr_no=0;

   int valid = 0;

   while(valid == 0){
         if(recvfrom(socket_desc,c_pkt,sizeof(*c_pkt),0,
                     (struct sockaddr *)&client_addr,&client_addr_size)<0){
                         printf("\nRecv error.\n");
                         return -1;
         }
         printf("\nRecved control pkt\n");
         valid = checkSumControl(c_pkt);
         ack->ack=1;
         ack->checkSum = createSumACK(ack);
         if(sendto(socket_desc,ack,sizeof(*ack),0,
                     (struct sockaddr *)&client_addr,client_addr_size)<0){
                         printf("\nError while sending akc to client.");
                         return -1;
         }
   }

   printf("\nSent control pkt ack.\n");
   return 1;

}

int comunicate(int socket_desc,struct sockaddr_in client_addr,int client_addr_size){
   struct data_pkt data;
   struct data_pkt * data1 = &data;
   memset(data1->msg,0,100);

   struct ack_pkt ackPKT;
   struct ack_pkt * ack = &ackPKT;

   if(recvfrom(socket_desc,data1,sizeof(*data1),0,
         (struct sockaddr *)&client_addr,&client_addr_size)<0){
                     printf("\nRecv error.\n");
                     return -1;
   }
   int valid = checkSum(data1);
   printf("\nMsg from client: %s",data1->msg);

   if(valid == 1){
         ack->ack=1;
         printf("\nData pkt is valid\n");
   }
   else{
         ack->ack=0;
         printf("\nData pkt is not valid\n");
   }
   ack->checkSum = createSumACK(ack);
   char c;
   scanf("%c",&c);

   if(sendto(socket_desc,ack,sizeof(*ack),0,
         (struct sockaddr *)&client_addr,client_addr_size)<0){
             printf("\nError while sending ack to client.");
             return -1;
   }


   if(sendto(socket_desc,ack,sizeof(*ack),0,
         (struct sockaddr *)&client_addr,client_addr_size)<0){
             printf("\nError while sending ack to client.");
             return -1;
   }

}

int comunicatePkt(int socket_desc,struct sockaddr_in client_addr,int client_addr_size,int
sr_no){
   struct data_pkt data;
   struct data_pkt * data1 = &data;
   memset(data1->msg,0,100);

   struct ack_pkt ackPKT;
   struct ack_pkt * ack = &ackPKT;

   if(recvfrom(socket_desc,data1,sizeof(*data1),0,
         (struct sockaddr *)&client_addr,&client_addr_size)<0){
             printf("\nRecv error.\n");
             return -1;
   }
   int valid = checkSum(data1);
   printf("\nchecksum: %ld",createSum(data1));
   printf("\nMsg from client: %s",data1->msg);

   if(valid == 1){
         if(data1->sr_no != sr_no) return -1;
         ack->ack=1;

         char * filename = "get.txt";
         FILE *fp = fopen(filename,"a");

         fprintf(fp,"%s",data1->msg);
         printf("\nmessage: %s",data1->msg);

         printf("\nData pkt is valid\n");
   }
   else{
         ack->ack=-1;
         printf("\nData pkt is not valid\n");
   }
   ack->checkSum = createSumACK(ack);

   if(sendto(socket_desc,ack,sizeof(*ack),0,
         (struct sockaddr *)&client_addr,client_addr_size)<0){
             printf("\nError while sending ack to client.");
             return -1;
   }
   return ack->ack;
}

int main(){
   int socket_desc;
   struct sockaddr_in server_addr,client_addr;
   int client_addr_size = sizeof(client_addr);

   socket_desc = socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP);
   if(socket_desc < 0){
         printf("\nError while creating socket.");
         return -1;
   }
   printf("\nCreated socket.");

   p
   server_addr.sin_family = AF_INET;
   server_addr.sin_port = htons(PORT);
   server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");

   if(bind(socket_desc,(struct sockaddr *)&server_addr,sizeof(server_addr))<0){
         printf("\nBinding error");
         return -1;
   }
   printf("\nBinding of socket.");

   printf("\nNow listening...");

   struct control_pkt pkt;


   int handshakeResult = shakeHand(socket_desc,client_addr,client_addr_size,&pkt);
   while(handshakeResult==-1){
         handshakeResult=shakeHand(socket_desc,client_addr,client_addr_size,&pkt);
         return -1;
   }

   printf("\nPackets : %d\n",pkt.no_of_pkt);

   for(int i=0;i<3;i++){
         int result = comunicatePkt(socket_desc,client_addr,client_addr_size,i);
         while(result == -1){
             result = comunicatePkt(socket_desc,client_addr,client_addr_size,i);
         }
   }
   return 1;
}
```

Server Screenshot :



```
saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$ gcc server.c -o server
saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$ ./server

Created socket.
Binding of socket.
Now listening...
Recved control pkt

Sent control pkt ack.

Packets : 1

checksum: 82907
Msg from client: hello

message: hello

Data pkt is valid

checksum: 53620
Msg from client: hii

message: hii

Data pkt is valid

checksum: 51077
Msg from client: bye

message: bye

Data pkt is valid
saif@saif:~/Desktop/Computer Networks/Lab 03/Q3$ 
```