DCCN LAB 5

NAME: **MALOTH ADITYA**
ROLL NO.: **120CS0124**

Q.1

Client Code:

```
// Client side implementation of UDP client-server model
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define SA struct sockaddr

int main(){
    int sid;
    char c='Y';
    struct sockaddr_in server_address;
    int ser_len;
    server_address.sin_family = AF_INET;
    server_address.sin_addr.s_addr = inet_addr("127.0.0.1");
    server_address.sin_port = 6969;
    ser_len = sizeof(server_address);
    printf("Character sent: %c\nSeeking a signal from server\n\n",c);
    sid=socket(AF_INET,SOCK_DGRAM,0);
    sendto(sid,&c,1,0,(SA *)&server_address,ser_len);
    recvfrom(sid,&c,1,0,(SA *)&server_address,&ser_len);
    printf("Character received from server: %c\n\n",c);
    close(sid);
    return 0;
}
```

Server Code:

```
// Server side implementation of UDP client-server model
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define SA struct sockaddr

int main(){
    int sockfd;
```

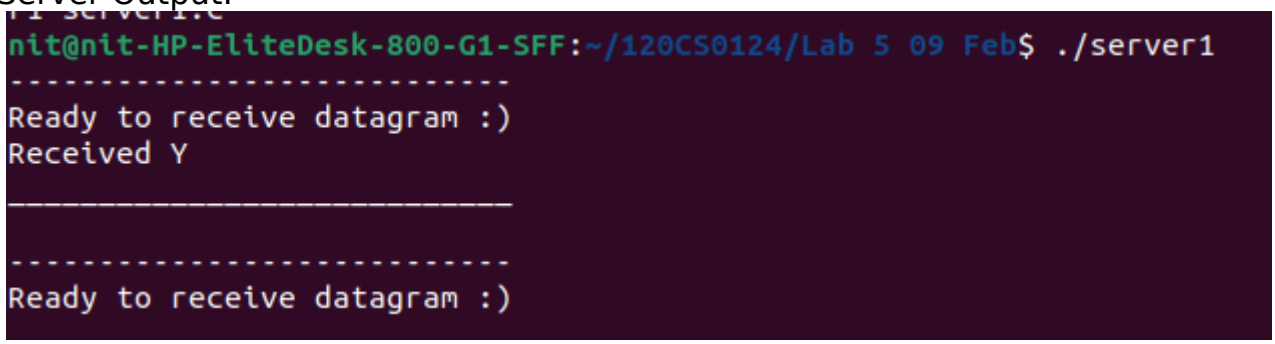
DCCN LAB 5

```
    char c;
    struct sockaddr_in servaddr, cliaddr;

    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = 6969;
    int ser_len = sizeof(servaddr);
    int cli_len = sizeof(cliaddr);
    sockfd = socket(AF_INET,SOCK_DGRAM,0);
    bind(sockfd,(SA *)&servaddr,ser_len);

    while(1){
        printf("--------------------------\n");
        printf("Ready to receive datagram :)\n");
        recvfrom(sockfd,&c,1,0,(SA *)&cliaddr,&cli_len);
        printf("Received %c\n",c);
        c='X';
        sendto(sockfd,&c,1,0,(SA *)&cliaddr,cli_len);
        printf("_____________________________\n\n");
    }
    close(sockfd);
    return 0;
}
```

Output:

Client Output:

Server Output:



DCCN LAB 5

Q.2

Client Code:

```
// Client side implementation of UDP client-server model
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define SA struct sockaddr

int main(){
	int sid,num1,num2,res;
	printf("Enter num1: ");
	scanf("%d",&num1);
	printf("Enter num2: ");
	scanf("%d",&num2);
	struct sockaddr_in server_address;
	int ser_len;
	server_address.sin_family = AF_INET;
	server_address.sin_addr.s_addr = inet_addr("127.0.0.1");
	server_address.sin_port = 6969;
	ser_len = sizeof(server_address);
	printf("Seeking a signal from server\n\n");
	sid=socket(AF_INET,SOCK_DGRAM,0);
	sendto(sid,&num1,sizeof(num1),0,(SA *)&server_address,ser_len);
	sendto(sid,&num2,sizeof(num2),0,(SA *)&server_address,ser_len);
	recvfrom(sid,&res,sizeof(res),0,(SA *)&server_address,&ser_len);
	printf("Sum received from server: %d\n",res);
	close(sid);
	return 0;
}
```

Server Code:

```
// Server side implementation of UDP client-server model
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define SA struct sockaddr

int main(){
```

DCCN LAB 5

```
        int sockfd,res,num;
        //const char hello = "Hello from server";
        struct sockaddr_in servaddr, cliaddr;

        servaddr.sin_family = AF_INET;
        servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
        servaddr.sin_port = 6969;
        int ser_len = sizeof(servaddr);
        int cli_len = sizeof(cliaddr);
        sockfd = socket(AF_INET,SOCK_DGRAM,0);
        bind(sockfd,(SA *)&servaddr,ser_len);

        while(1){
                printf("_____\n");
                printf("Ready to receive datagram :)\n");
                recvfrom(sockfd,&res,sizeof(res),0,(SA *)&cliaddr,&cli_len);
                recvfrom(sockfd,&num,sizeof(num),0,(SA *)&cliaddr,&cli_len);
                printf("Received num1: %d\tnum2: %d\n",res,num);
                res=res+num;
                sendto(sockfd,&res,sizeof(res),0,(SA *)&cliaddr,cli_len);
                printf("--------------------------\n\n");
        }
        close(sockfd);
        return 0;
}
```

Output:

Client Output:

```
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab 5 09 Feb$ ./client2
Enter num1: 23
Enter num2: 46
Seeking a signal from server

Sum received from server: 69
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab 5 09 Feb$
```

Server Output:

```
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab 5 09 Feb$ ./server2
_____
Ready to receive datagram :)
Received num1: 23       num2: 46
--------------------------


_____
Ready to receive datagram :)
```

DCCN LAB 5

Q.3

Client Code in UDP:

```
// Client side implementation of UDP client-server model
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define SA struct sockaddr

int main(){
	int sid,num1,num2,res;
	printf("Enter num1: ");
	scanf("%d",&num1);
	printf("Enter num2: ");
	scanf("%d",&num2);
	struct sockaddr_in server_address;
	int ser_len;
	server_address.sin_family = AF_INET;
	server_address.sin_addr.s_addr = inet_addr("127.0.0.1");
	server_address.sin_port = 6969;
	ser_len = sizeof(server_address);
	printf("\n\nSeeking a signal from server\n");
	sid=socket(AF_INET,SOCK_DGRAM,0);
	sendto(sid,&num1,sizeof(num1),0,(SA *)&server_address,ser_len);
	sendto(sid,&num2,sizeof(num2),0,(SA *)&server_address,ser_len);
	int diff,prod,quo;
	recvfrom(sid,&diff,4,0,(SA *)&server_address,&ser_len);
	recvfrom(sid,&prod,4,0,(SA *)&server_address,&ser_len);
	recvfrom(sid,&quo,4,0,(SA *)&server_address,&ser_len);
	//for(int i=0;i<3;i++) rans[i]=ntohl(ans[i]);
	printf("Heard something from server\n");
	printf("\nDifference: %d\nProduct: %d\nQuotient: %d\n",diff,prod,quo);
	if(quo==-1) printf("Since num2 is 0, couldn't perform division\n");
	close(sid);
	return 0;
}
```

Server Code in UDP:

```
// Server side implementation of UDP client-server model
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
//#include <string.h>
#include <sys/types.h>
```

DCCN LAB 5

```
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define SA struct sockaddr

#define PORT 6969
#define MAXLINE 1024

int main(){
	int sockfd,res,num;
	struct sockaddr_in servaddr, cliaddr;

	servaddr.sin_family = AF_INET;
	servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
	servaddr.sin_port = 6969;
	int ser_len = sizeof(servaddr);
	int cli_len = sizeof(cliaddr);
	sockfd = socket(AF_INET,SOCK_DGRAM,0);
	bind(sockfd,(SA *)&servaddr,ser_len);
	int diff,prod,quo;
	while(1){
		printf("_____\n");
		printf("Ready to receive datagram :)\n");
		recvfrom(sockfd,&res,sizeof(res),0,(SA *)&cliaddr,&cli_len);
		recvfrom(sockfd,&num,sizeof(num),0,(SA *)&cliaddr,&cli_len);
		printf("Received num1: %d\tnum2: %d\n",res,num);
		diff = res-num;
		prod = res*num;
		//for(int i=0;i<3;i++) sans[i]=htonl(ans[i]);
		if(num==0) quo=-1;
		else quo = res/num;
		printf("\nDifference: %d\nProduct: %d\nQuotient: %d\
n",diff,prod,quo);
		sendto(sockfd,&diff,4,0,(SA *)&cliaddr,cli_len);
		sendto(sockfd,&prod,4,0,(SA *)&cliaddr,cli_len);
		sendto(sockfd,&quo,4,0,(SA *)&cliaddr,cli_len);
		printf("---------------------------\n\n");
	}
	close(sockfd);
	return 0;
}
```

Output:
Client Output:

DCCN LAB 5

```
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab 5 09 Feb$ ./client3
Enter num1: 23
Enter num2: 46


Seeking a signal from server
Heard something from server

Difference: -23
Product: 1058
Quotient: 0
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab 5 09 Feb$
```

Server Output:

```
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab 5 09 Feb$ ./server3
____________________________
Ready to receive datagram :)
Received num1: 23       num2: 46

Difference: -23
Product: 1058
Quotient: 0
----------------------------


____________________________
Ready to receive datagram :)
```

Q.3

Client Code in TCP:

```
// Client side implementation of TCP client-server model
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <arpa/inet.h>
int main(){
    int sid,num1,num2;
        printf("Enter num1: ");
        scanf("%d",&num1);
        printf("Enter num2: ");
        scanf("%d",&num2);
    struct sockaddr_in server_address;
    int server_addlen;
    server_address.sin_family=AF_INET;
```

DCCN LAB 5

```
    server_address.sin_addr.s_addr=inet_addr("127.0.0.1");
    server_address.sin_port=5080;
    server_addlen=sizeof(server_address);
    sid=socket(AF_INET,SOCK_STREAM,0);
    connect(sid,(struct sockaddr *)&server_address,server_addlen);
    printf("Sending data to server\n");
    write(sid,&num1,sizeof(int));
    write(sid,&num2,sizeof(int));

    int diff,prod,quo,ans[3];
    read(sid,&diff,sizeof(int));
    read(sid,&prod,sizeof(int));
    read(sid,&quo,sizeof(int));
    printf("\nDifference: %d\nProduct: %d\nQuotient: %d\n",diff,prod,quo);
    if(quo==-1) printf("Since num2 is 0, couldn't perform division\n");
    close(sid);
    return(0);
}
```

Server Code in TCP:

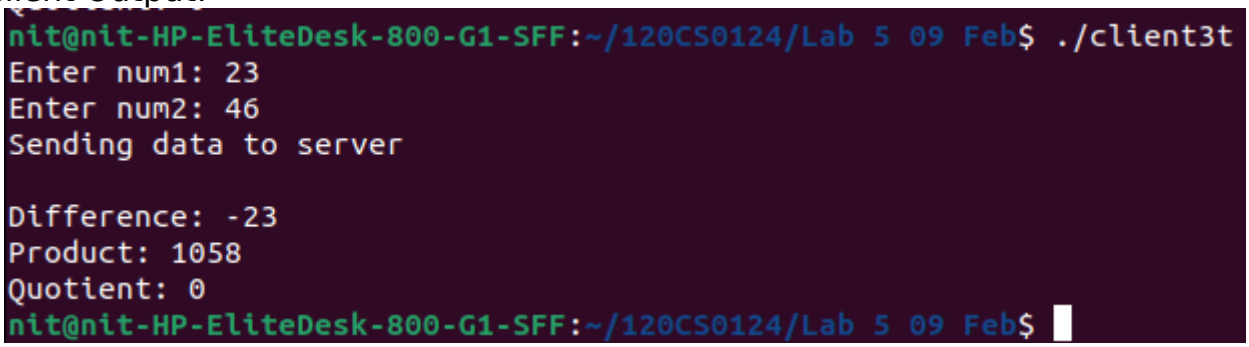
```
// Server side implementation of TCP client-server model
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <arpa/inet.h>
int main(){
    int serid,sessid;
    int res,num;
    struct sockaddr_in server_address,client_address;
    int server_addlen,client_addlen;
    server_address.sin_family=AF_INET;
    server_address.sin_addr.s_addr=inet_addr("127.0.0.1");
    server_address.sin_port=5080;
    server_addlen=sizeof(server_address);
    client_addlen=sizeof(client_addlen);
    serid=socket(AF_INET,SOCK_STREAM,0);
    bind(serid,(struct sockaddr*)&server_address,server_addlen);
    listen(serid,10);
    while(1){
        printf("_____\n");
       printf("Server is ready to accept ......\n");
       sessid=accept(serid,(struct sockaddr *)&client_address,&client_addlen);
       read(sessid,&res,sizeof(int));
       read(sessid,&num,sizeof(int));
       printf("Received num1: %d\tnum2: %d\n",res,num);
       int prod,diff,quo;
```

DCCN LAB 5

```
        diff = res-num;
         prod = res*num;
         if(num==0) quo=-1;
         else quo = res/num;
         printf("\nDifference: %d\nProduct: %d\nQuotient: %d\n",diff,prod,quo);
         write(sessid,&diff,sizeof(int));
         write(sessid,&prod,sizeof(int));
         write(sessid,&quo,sizeof(int));
         printf("---------------------------------\n\n");
        close(sessid);
    }
    return(0);
}
```

Output:

Client Output:



Server Output: