

Exercise 7

Codes:

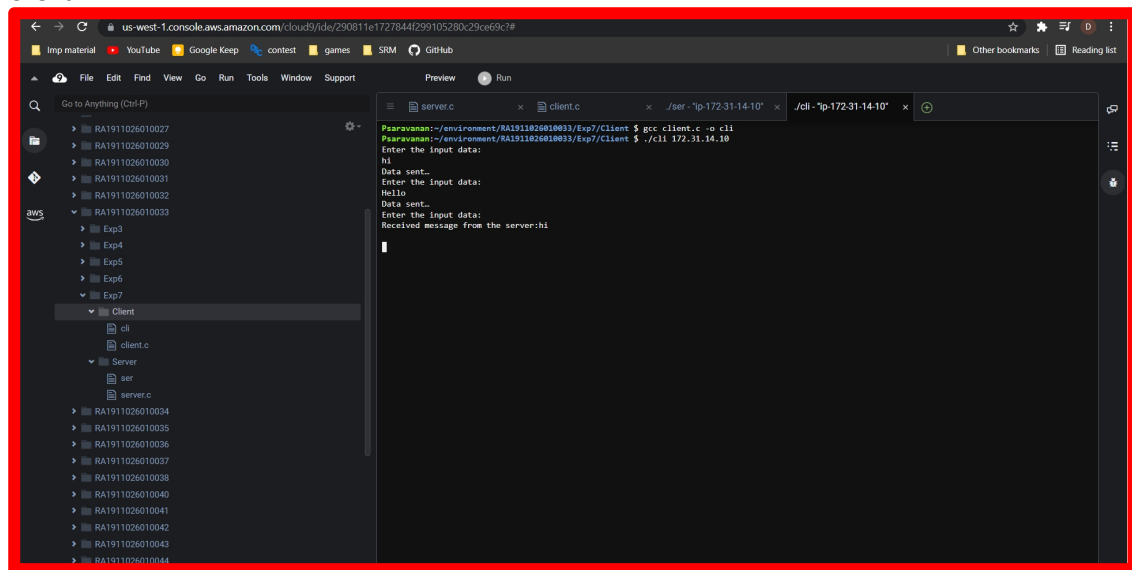
Server :

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<unistd.h>
#include<netdb.h>
#include<arpa/inet.h>
#include<netinet/in.h>
int main(int argc, char *argv[])
{
    int ad, sd;
    struct sockaddr_in servaddr, cliaddr;
    socklen_t servlen, clilen;
    char buff[1000], buff1[1000];
    pid_t cpid;
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family=AF_INET;
    servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
    servaddr.sin_port=htons(5500);
    sd=socket(AF_INET, SOCK_STREAM, 0);
    bind(sd, (struct sockaddr*)&servaddr, sizeof(servaddr));
    listen(sd, 5);
    printf("%s\n", "Server is running.....");
    ad=accept(sd, (struct sockaddr*)&cliaddr, &clilen);
    cpid=fork();
    if(cpid==0)
    {
        while(1)
        {
            bzero(&buff, sizeof(buff));
            recv(ad, buff, sizeof(buff), 0);
            printf("Received message from the client:%s\n", buff);
        }
    }
    else
    {
        while(1)
        {
            bzero(&buff1, sizeof(buff1));
            printf("%s\n", "Enter the input data:");
            fgets(buff1, 10000, stdin);
            send(ad, buff1, strlen(buff1)+1, 0);
            printf("%s\n", "Data sent...");
        }
    }
    return 0;
}
```

Client:

```
#include<sys/socket.h>
#include<sys/types.h>
#include<stdio.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<netdb.h>
#include<netinet/in.h>
int main(int argc,char *argv[])
{
    int sd,cd;
    struct sockaddr_in servaddr,cliaddr;
    socklen_t servlen,clilen;
    char buff[1000],buff1[1000];
    pid_t cpid;
    bzero(&servaddr,sizeof(servaddr));
    servaddr.sin_family=AF_INET;
    servaddr.sin_addr.s_addr=inet_addr(argv[1]);
    servaddr.sin_port=htons(5500);
    sd=socket(AF_INET,SOCK_STREAM,0);
    cd=connect(sd,(struct sockaddr*)&servaddr,sizeof(servaddr));
    cpid=fork();
    if(cpid==0)
    {
        while(1)
        {
            bzero(&buff,sizeof(buff));
            printf("%s\n","Enter the input data:");
            fgets(buff,10000,stdin);
            send(sd,buff,strlen(buff)+1,0);
            printf("%s\n","Data sent...");
        }
    }
    else
    {
        while(1)
        {
            bzero(&buff1,sizeof(buff1));
            recv(sd,buff1,sizeof(buff1),0);
            printf("Received message from the server:%s\n",buff1);
        }
    }
    return 0;
}
```

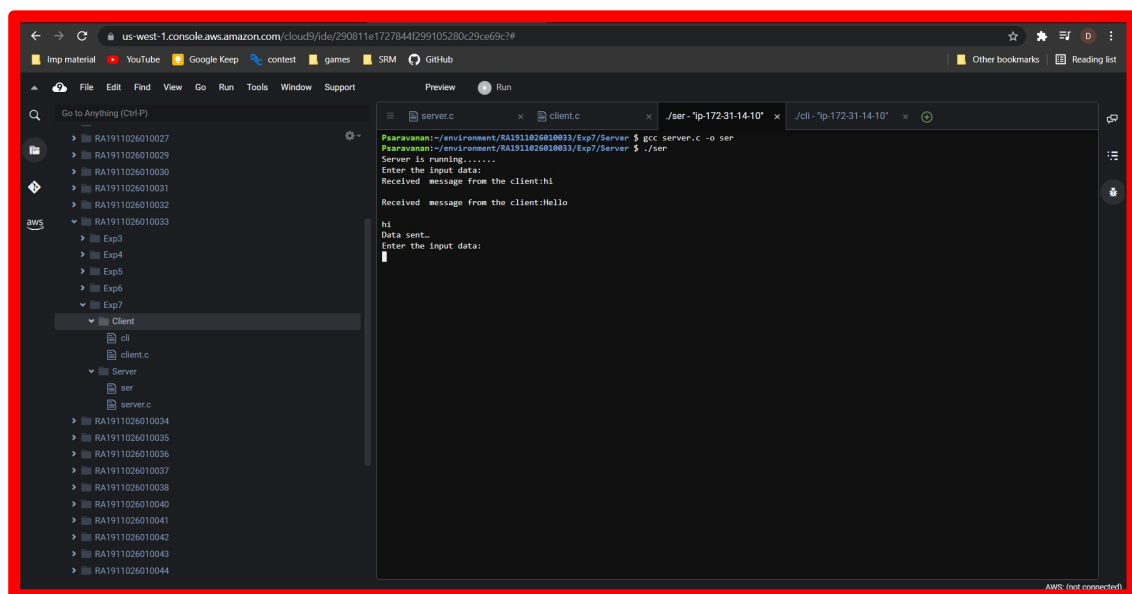
Ouput :
Client:



The screenshot shows the AWS Cloud9 IDE interface. The left sidebar displays a file explorer with a project structure including 'Client' and 'Server' folders. The main terminal window is titled 'Preview' and shows the output of the client program. The output text is as follows:

```
Pearavanan:~/environment/RA1911026010033/Exp7/Client $ gcc client.c -o cli
Pearavanan:~/environment/RA1911026010033/Exp7/Client $ ./cli 172.31.14.10
Enter the input data:
hi
Data sent..
Enter the input data:
Hello
Data sent..
Enter the input data:
Received message from the server:hi
```

Server:



The screenshot shows the AWS Cloud9 IDE interface. The left sidebar displays a file explorer with a project structure including 'Client' and 'Server' folders. The main terminal window is titled 'Preview' and shows the output of the server program. The output text is as follows:

```
Pearavanan:~/environment/RA1911026010033/Exp7/Server $ gcc server.c -o ser
Pearavanan:~/environment/RA1911026010033/Exp7/Server $ ./ser
Server is running.....
Enter the input data:
Received message from the client:hi
Received message from the client:Hello

hi
Data sent..
Enter the input data:
```

AWS (not connected)