

**Server:**

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

struct sockaddr_in s;
struct sockaddr_in cl;

int main()
{

int MSG_CONFIRM = 0;
int fd = socket(AF_INET, SOCK_DGRAM, 0), port = 1051;
if(fd<0){
printf("Error! Opening Socket");
return 0;
}
printf("connected successfully\n");
char buffer[250];
memset(&cl, 0 ,sizeof(cl));
memset(&s, 0, sizeof(s));
s.sin_family = AF_INET;
s.sin_addr.s_addr = INADDR_ANY;
s.sin_port = htons(port);

if( bind(fd, (const struct sockaddr *)&s, sizeof(s) ) >0){
printf("Error! binding failed");
return 0;
}
printf("Bind successful\n"); unsigned int len = sizeof(cl) , n;
printf("Server ready to chat\n");

while(1){ bzero(buffer, 250);
n= recvfrom(fd, (char *)buffer, 250, MSG_WAITALL, (struct sockaddr *)&cl , &len);
buffer[n]='\0';
if(n<0){
printf("Error reading");
return 0;
}
printf("Client : %s \n", buffer);
bzero(buffer, 250);
printf("You [server]: ");
fgets(buffer, 250, stdin);
sendto(fd, (const char *)buffer, strlen(buffer), MSG_CONFIRM, (const struct sockaddr *)&cl,
len);

if (strncmp("END", buffer, 3) == 0) {
printf("Server Terminating...\n");
break;
}
```

```

}
}
printf("Server Terminated\n");
close(fd);
return 0;

}

```

## **Client:**

```

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

struct sockaddr_in s;

int main ()
{
    int MSG_CONFIRM = 0;
    int fd = socket(AF_INET, SOCK_DGRAM, 0), port = 1051;
    if(fd<0){
        printf("Error! Opening Socket");
        return 0;
    }
    printf("connected successfully\n");
    memset(&s, 0, sizeof(s));
    char buffer[250];
    s.sin_family = AF_INET;
    s.sin_addr.s_addr = INADDR_ANY; s.sin_port = htons(port);

    if( bind(fd, (const struct sockaddr *)&s, sizeof(s) ) >0){
        printf("Error! binding failed");
        return 0;
    }

    int n;
    unsigned int len = sizeof(s);
    printf("Client ready to chat\n");

    while(1){

        bzero(buffer, 250); printf("You [client]: ");
        fgets(buffer, 250, stdin);
        sendto(fd, (const char *)buffer, strlen(buffer), MSG_CONFIRM, (const struct sockaddr*)&s,
            sizeof(s));
        bzero(buffer, 250);
        n= recvfrom(fd, (char *)buffer, 250, MSG_WAITALL, (struct sockaddr *)&s , &len);
        buffer[n]='\0';
        if(n<0){
            printf("Error reading");
            return 0;
        }
        printf("Server : %s \n", buffer);
        if (strncmp("END", buffer, 3) == 0) {
            printf("Client Terminating...\n");
            break;
        }
    }
}

```

```

}
printf("Client Terminated\n");
close(fd);
return 0;
}

```

## OUTPUT:-

The screenshot shows the AWS Cloud9 IDE interface. The left sidebar displays a file explorer with a project structure including folders for RA1911027010067 through RA1911027010068, and subfolders for exp1-tcp, exp2-udp, and exp3-timeserver. The main editor area shows the client78.c file. The terminal output is as follows:

```

RA1911027010076:~/environment/RA1911027010078/exp2-udp $ cc -o client client78.c
RA1911027010076:~/environment/RA1911027010078/exp2-udp $ ./client
connected successfully
Client ready to chat
You [client]: hii
Server : hello i am server

You [client]: ok
Server : END

Client Terminating...
Client Terminated
RA1911027010076:~/environment/RA1911027010078/exp2-udp $

```

The screenshot shows the AWS Cloud9 IDE interface. The left sidebar displays a file explorer with a project structure including folders for RA1911027010067 through RA1911027010068, and subfolders for exp1-tcp, exp2-udp, and exp3-timeserver. The main editor area shows the server78.c file. The terminal output is as follows:

```

RA1911027010076:~/environment/RA1911027010078/exp2-udp $ cc -o server server.c
cc: error: server.c: No such file or directory
cc: fatal error: no input files
compilation terminated.
RA1911027010076:~/environment/RA1911027010078/exp2-udp $ cc -o server server78.c
RA1911027010076:~/environment/RA1911027010078/exp2-udp $ ./server
connected successfully
Bind successful
Server ready to chat
Client : hii

You [server]: hello i am server
Client : ok

You [server]: END
Server Terminating...
Server Terminated
RA1911027010076:~/environment/RA1911027010078/exp2-udp $

```