Exercise 8

Codes :

Server:

```
        #include <arpa/inet.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int main(int argc, char *argv[])
{
    int sd, ad, size;
    struct sockaddr_in servaddr, cliaddr;
    socklen_t clilen;
    clilen = sizeof(cliaddr);
    struct stat x;
    char buff[100], file[10000];
    FILE *fp;

    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(1500);
    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);
    while (1)
    {
        bzero(buff, sizeof(buff));
        bzero(file, sizeof(file));
        recv(ad, buff, sizeof(buff), 0);
        fp = fopen(buff, "r");
        stat(buff, &x);
        size = x.st_size;
        fread(file, sizeof(file), 1, fp);
        send(ad, file, sizeof(file), 0);
    }
}
```

Client:

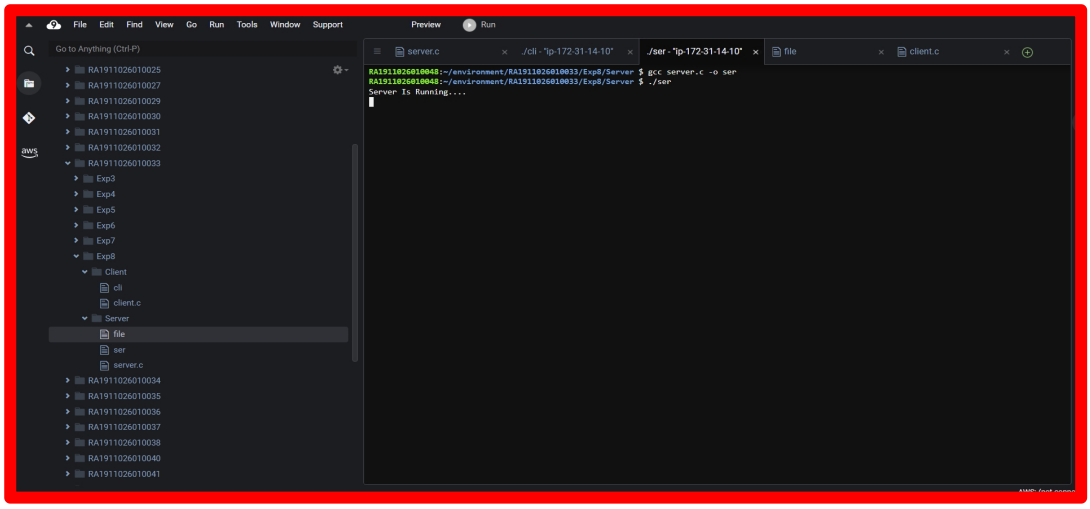
```
    #include <arpa/inet.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdio.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <unistd.h>
int main(int argc, char *argv[])
{
    int sd, cd;
    struct sockaddr_in servaddr, cliaddr;
    socklen_t clilen;
    char buff[100], file[10000];
    struct hostent *h;

    h = gethostbyname(argv[1]);
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = h->h_addrtype;
    memcpy((char *)&servaddr.sin_addr.s_addr, h->h_addr_list[0], h->h_length);
    servaddr.sin_port = htons(1500);
    sd = socket(AF_INET, SOCK_STREAM, 0);
    cd = connect(sd, (struct sockaddr *)&servaddr, sizeof(servaddr));
    while (1)
    {
        printf("%s\n", "Enter the File Name :");
        scanf("%s", buff);
        send(sd, buff, strlen(buff) + 1, 0);
        printf("%s\n", "File Output :");
        recv(sd, file, sizeof(file), 0);
        printf("%s", file);
    }
    return 0;
}
```

Content of file in the server:

File name is :file

```
<sys/types.h>
<sys/uio.h>
<errno.h>
<sys/ioctl.h>
<fcntl.h>
<sys/socket.h>
<netdb.h>
<netinet/in_systm.h>
<netinet/ip_icmp.h>
<netinet/udp.h>
<netinet/ip.h>
<netinet/in.h>
<arpa/inet.h>
<arpa/nameser.h>
<resolv.h>
<net/if.h>
<strings.h>
```

Output
Server



Client: