```
function main
    if started as server
        srvr()
    else if started as client
        clnt()
    else
        print proper usage


function srvr
    initialize variables for fork, sockets, address structs

    open tcp socket
    if opening tcp socket failed
        print error and exit

    bind tcp socket to waiting port
    if could not bind socket
        print error and exit

    listen for new connections

    infinite loop
        block until a new connection is accepted
        if there is an error accepting
            continue

        fork

        if there is a fork error
            clean up socket

        parent
            continue

        child
            serveClient(clientSocket, clientAddress)
            clean up socket


function serveClient(clientSocket, clientAddress)
    initialize buffer
    read all data from clientSocket into buffer
```

```
    if first character of buffer is not STX
        return

    if second character is G
        respondGETRequest(clientAddress, buffer)

    else if second character is S
        respondSENDRequest(clientAddress, buffer)

    else
        error


function respondGETRequest(clientAddress, buffer)
    initialize variables for socket, address, and file
    read filename from buffer
    open the file

    if the file wasnt openned
        error
        return

    create a new TCP socket
    if the socket could not be created
        error
        return

    connect to clientAddress on data port using new TCP socket
    if could not connect
        close socket and file
        error
        return

    write file into the new TCP socket

    close file and socket


function respondSENDRequest(clientAddress, buffer)
    initialize variables for address and file

    read filename and file length from buffer

    create a new tcp socket
```

```
    if the socket could not be created
        error
        return

    connect to clientAddress on data port using new TCP socket

    open the requested file

    send ack packet on data socket
    send file over data socket

    close file
    close socket


function clnt
    grab filename and serverAddress from command line arguments

    if GET specified
        handleGET(serverAddress, filename)

    else if SEND specified
        handleSEND(serverAddress, filename)

    else
        print proper usage of program


function handleGET(serverAddress, filename)
    initialize variables for file, socket and address

    create a new TCP socket
    if socket could not be openned
        error
        return

    create a new address struct from serverAddress
    if new struct could not be made
        error
        return

    connect to address
    if could not connect
        error
```

```
        return

    send the get request for filename
    close the socket

    create a new TCP socket
    if socket could not be openned
        error
        return

    bind socket to data port
    if bind failed
        error
        return

    listen for new connection

    accept the new connection
    if failed to accept
        error
        return

    close the listen socket

    open a the file
    read all the data into the file
    close the file
    close the socket


function handleSEND(serverAddress, filename)
    initialize socket and address variables

    create a tcp socket
    if failed to create socket
        error
        return

    create address struct from serverAddress
    if failed to create
        error
        return

    conenct to server
```

```
if failed to connect
    close socket
    error
    return

open the file
if failed to open
    close socket
    error
    return

send send request for file to server
close socket

create new TCP socket
if create failed
    close file
    error
    return

bind socket to data port
if bind failed
    close file
    close socket
    error
    return

listen for new connection

block until new connection accepted
if new accept failed
    close file
    close socket
    error
    return

close listen socket

read the data in the data port into file

close the file
close the socket
```