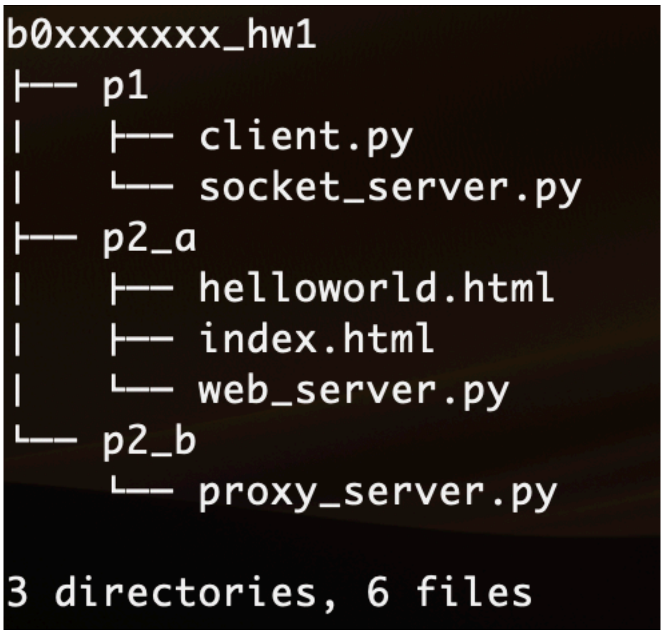
Homework1

Due date: 2021/10/27 23:59

Homework Policy:(Read before you start to work)

1. 作業請勿抄襲，如果被發現，作業以零分計算
2. 如果作業上遇到困難可以討論，但是手寫和程式碼的部分必須是你自己完成，並且請在作業的 pdf 檔註明討論的同學姓名及學號
3. 程式作業請於期限內至 ceiba 作業區上傳，格式為 zip 檔，解壓縮後應恰為一個以學號為名的資料夾，資料夾內的檔案如下所示



```
b0xxxxxxx_hw1
├── p1
|   ├── client.py
|   └── socket_server.py
├── p2_a
|   ├── helloworld.html
|   ├── index.html
|   └── web_server.py
└── p2_b
    └── proxy_server.py

3 directories, 6 files
```

Figure 1: folder tree

4. 逾期繳交一天內，分數 $\times\frac{2}{3}$，超過一天未滿兩天，分數 $\times\frac{1}{3}$，超過兩天則不予計分，請務必盡早開始，並努力完成。
5. 如有任何問題歡迎來信，並請在郵件的標題註明課程。
   範例:[2021ICN] 作業一問題
   信箱: 連潔琳 R08942159@ntu.edu.tw

Problems:

1. **Socket Programming - TCP** – 40%
   We are going to construct a TCP server and client system. Fig 2 describes the messages sent between client and server.

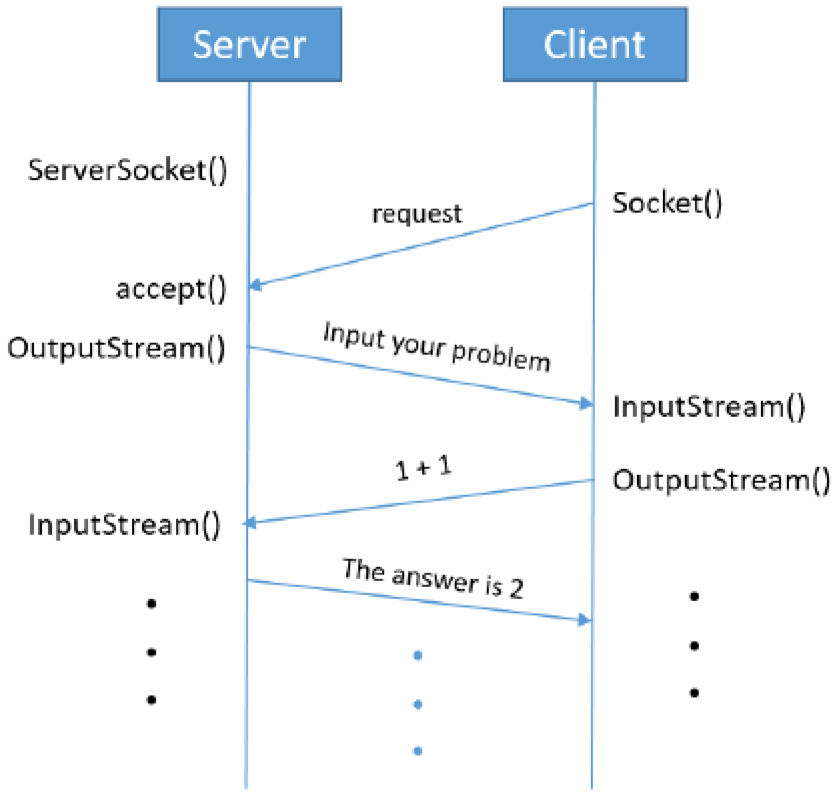


Figure 2: Message flow between client and server

Please refer to the following steps:

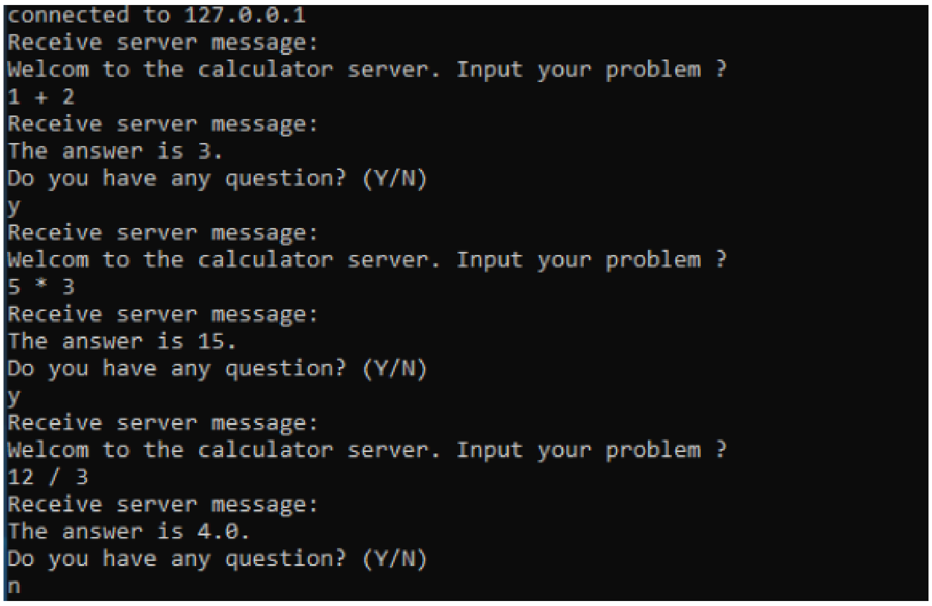
(a) Read the source of socket_server.py (works on python3 environment), try your best to understand it.

(b) Add some code at "TODO" part of socket_server.py. Let the server to work as a calculator that support basic "+","-","×","÷" operations. DON'T need to handle error input and Split the number and operators with one space.

(c) Write your own socket_client.py such that it can (1)Read the message sent from server and (2)Send the user-input message to the server to check if the server works as you expect. The message flow between client and server is shown in Fig 2.

(d) Compile and run the socket_server.py first, and then run your socket_client.py.

(e) Test it under your local machine. If you do it correct, the execution and output of client.py should be like Fig 3.



```
connected to 127.0.0.1
Receive server message:
Welcom to the calculator server. Input your problem ?
1 + 2
Receive server message:
The answer is 3.
Do you have any question? (Y/N)
y
Receive server message:
Welcom to the calculator server. Input your problem ?
5 * 3
Receive server message:
The answer is 15.
Do you have any question? (Y/N)
y
Receive server message:
Welcom to the calculator server. Input your problem ?
12 / 3
Receive server message:
The answer is 4.0.
Do you have any question? (Y/N)
n
```

Figure 3: Connect to server on local machine(127.0.0.1)

(f) It really works! Try to run your socket_client.py and connect to TA's computer with TA's IP address 140.112.42.108 and port 7777.

(g) (Bonus–5%)You can also implement other math function. For example, matrix multiplication, integral, and Laplace's transform. Each function you add, you get extra 1 point in this problem. However, the existing math libraries are not allowed. You should write these steps by your own and specify how to call these functions in README!

(h) (Grading)Each operator will take 7 points. You can get the less 12 points by implementing the socket. We will test your client with your server and our server both. Set the IP as '127.0.0.1' and Port '8888' first. Please put your source code(including socket_server.py and client.py) under [student_id]/src/p1. If you have some other function done in (g), you should also put README.md under [student_id]/src/p1.

2. **Web Server** -25%(Chap 2 Programming Assignment#1,#4 in textbook)
In this assignment, you will develop a simple Web server in python that is capable of processing simple request. There should be at least two html files. One is [index.html] and another is [helloworld.html]. You can access the two html files through the Web server. Also, you can access the [helloworld.html] through the [index.html].

Please refer to the following steps:

(a) Run the Web server on the local machine and create a connection socket

(b) Receive the HTTP request from this connection when contacted by a client(browser)

(c) Parse the request to determine the specific file being requested

(d) Get the requested file from the server's file system

(e) Create an HTTP response message consisting of the requested file preceded by header lines

(f) If the client send HTTP request for the other pages through the homepage, the web server will go back to the step (b)

(g) Send the response over the TCP connection to the requesting browser. If a browser requests a file that is not present in your server, your server should return a "404 Not Found" error message

The message flow between client and web browser and web server is shown in Fig 4.

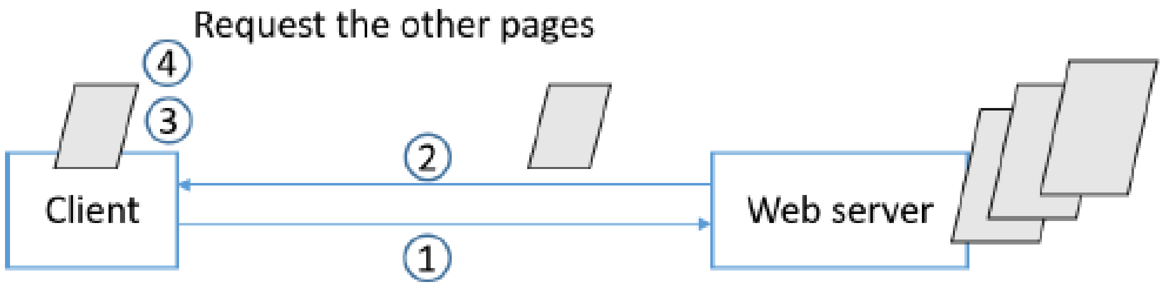


Figure 4: Message flow between the client and the web server

Read the source code web_server.py, and try your best to understand it. Add some codes to the "TODO" part to complete the function of web server. Put html files under the directory that web server is running, and get the file through the browser(127.0.0.1:8888/FILENAME).
Get your IP address of your web server, and try to connect to the web server by another device instead of the local machine such as your cellphone. Check if you can get the file in another device through browser.
Finally, put your source code(web_server.py) and html files(index.html/helloworld.html) under [student_id]/src/p2_a and set the IP as '127.0.0.1' and Port '8888'.

3. **Proxy Server** -35%
In this assignment, you will develop a Web proxy. The message is shown if Fig 5. Followings are the things that you have to do.

(a) Create a socket on the proxy server and receive data(request) from the client

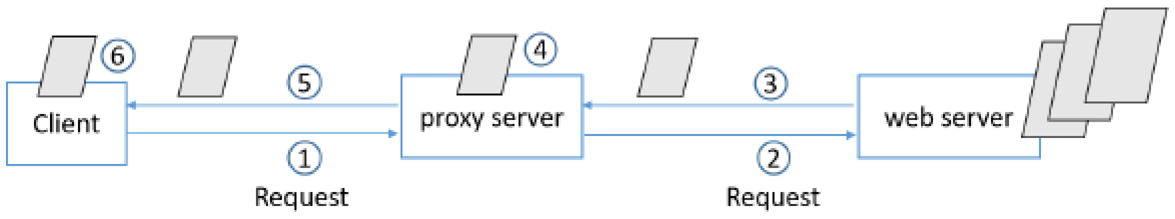


Figure 5: Message flow among Client-Proxy-Server

(b) Parse the request to determine the specific file being requested

(c) Find in the local whether there is the file that requested by the client. If yes, send the file back as you did in the p2_a, but if not, go to the step(d)

(d) Create another socket on the proxy server and connect to the web server that you did in the p2_a

(e) Ask the web server for the file requested by the client

(f) Receive and read the response and send it to the client. Also, save the file in the local as cache

(g) If the file is not found, return an error message

The skeleton code is provided to you in the attached file with file name proxy_server.py. Your job is to complete the code in the "TODO" part, and then test it by having different browsers request Web objects via your proxy. Finally, out your source code(proxy_server.py) under [student_id]/src/p2_b and set the IP as '127.0.0.1' and Port '7777' first.