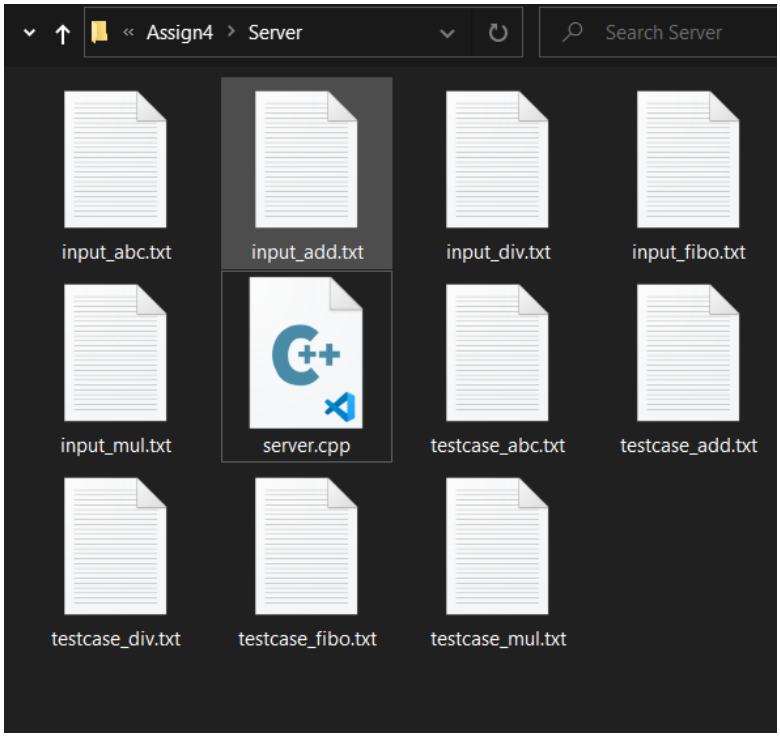
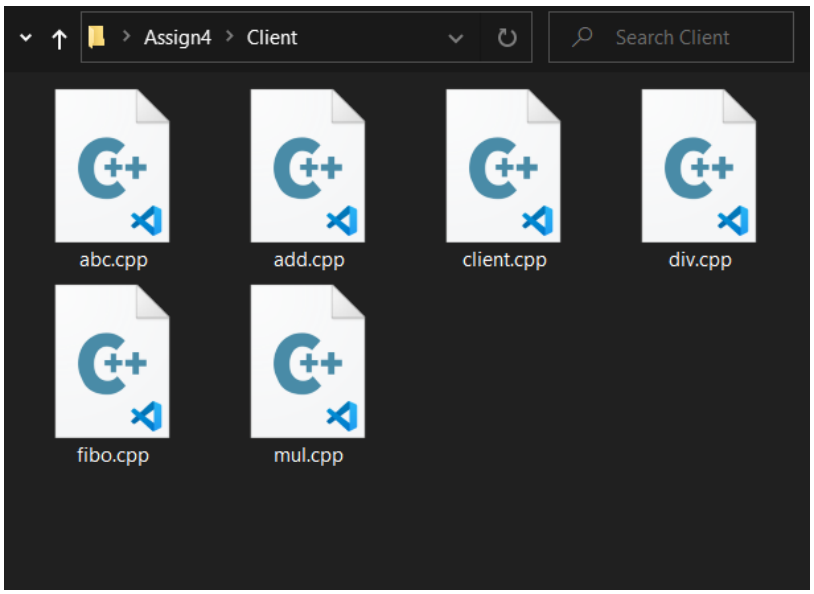
Folder content:

- ✓ Server:



- ✓ Client:



N.B.:

1. The ***server.cpp*** / ***server.c*** and ***client.cpp*** / ***client.c*** file will be written by the students.
2. All other c/cpp files in client's folder will be used for the testing purpose.
3. For all test c/cpp file there will be a testcase file and a input file (if needed) at the server side (these file will be created and added by the TA, one of two such file maybe shared with you).
4. The naming convention for input file will be input _<testprog_name_without_extension>.txt and testcase file name will be testcase_<testprog_name_without_extension>.txt,

- For example if test program name is add.cpp then
    - Input file name: input_add.txt
    - Testcase file name: testcase_add.txt

5. You can use input output redirection for providing input to the test program and to store the output of the program in a text file (naming convention output_add.txt (for previous example)). Then you need to compare output_add.txt with testcase_add.txt.
6. Let us see a demo test program:

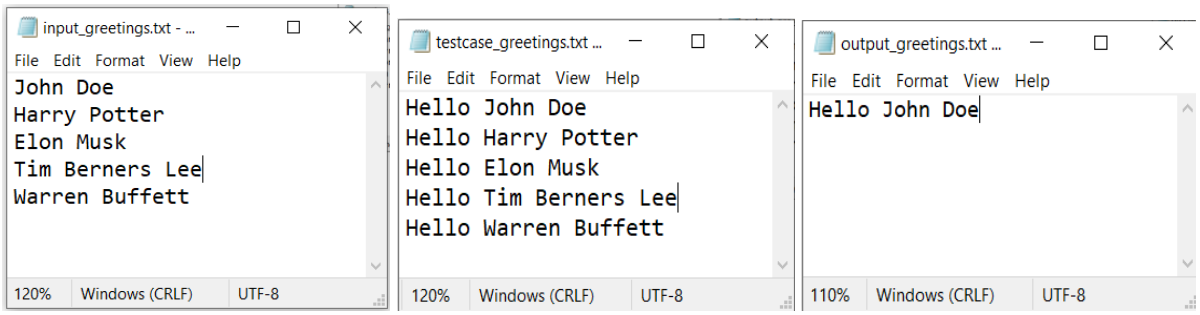
greetings.cpp

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[])
{
    string name;
    getline(cin, name);
    cout << "Hello " << name << endl;
    return 0;
}
```

N.B.: ***All input*** in a ***test program*** will be read using ***cin or scanf*** function. Student can use input redirection so that the input comes from the input_greetings.txt file.

Eg. `g++ greetings.cpp && ./a.out <0 input_greetings.txt >1 output_greetings.txt`

If you run the above command the program will ***read*** the first line ***from input_greetings.txt*** and ***store*** the output into ***output_greetings.txt.***



7. ***Please Note : input_greetings.txt*** may have multiple lines of input but above code will read only one line (refers to above picture). You have to write the server.c(/.cpp) file such that given the exact code (greetings.cpp) it should produce output for all input testcases (which will be stored in one txt. Each line should be consider as a different test input).

   → ***output_greetings.txt*** file should look like this after successful execution.