Report

How to run the code

- This number-guessing game consists 2 python files: `TCPServer.py` and `TCPClient.py`.
- Start the server before running the client program. First, run `TCPServer.py` by entering `python3 TCPServer.py` in the terminal(we assume that the testing environment is python3); and then run `python3 TCPClient.py` in another terminal. After seeing 'Welcome to the guessing game!' on the client side, we may start the game.

How to play

- The client generates hints on the terminal to guide you through the game. All you need to do is to enter the number you guess.
- After each round of the game, you'll have the option to start another round of the game. type y for yes, and other things for no. Note that only 'y' is recognized as yes.

How the two processes interact

- The output of a game is basically like this:

```
zhengry22@zhengruiyang:/mnt/d/计算机网络原理/Homework_1$ python3 TCPClient.py
Welcome to the guessing game!
Please enter your guess
1
From server: Smaller than answer! Try again!
Please enter your guess
100
From server: Greater than answer! Try again!
Please enter your guess
6
From server: Congratulations! You win with 3 guesses! Would you like to start a new game? [y for yes / other input for no]
y
Please enter your guess
1
From server: Smaller than answer! Try again!
Please enter your guess
100
From server: Greater than answer! Try again!
Please enter your guess
91
From server: Congratulations! You win with 3 guesses! Would you like to start a new game? [y for yes / other input for no]
n
```

  - During each loop, the client requires an input as our guess.
  - If the input is not the answer, the server sends an ack with the hint in it, and prepares for the next reception.
  - The client trys until it gets the correct answer. Afterwards, if it decides to continue the game, input 'y' and the connection remains. Else, the link terminates via the `close()` function.

- The way client and server interacts can be shown in the following picture: