# Homework 1

B00401062 羅文斌

## Program structure

The Python program is organized into the following sections: (1) Define constants, (2) Read the configuration file, (3) Socket connection, (4) Send configuration message to Irssi, and (5) Handle input cases.

1. Define constants. Constants, including the host name with which to build connection, the nickname, the true identity, and the real name for Irssi are defined.
2. Socket creation. Socket is created with IPv4 and TCP protocols and connected to the designated host name at some port number.
3. Send configuration messages to Irssi. Configuration messages including NICK (for nickname), USER (for user identity), and JOIN (for joining a channel) are required to enter a chat room. And then a “hello” message is sent.
4. Handle input cases. Input cases are handle in an infinite while loop. The process blocks to read messages from the socket and acts according to the keywords in the input line. Line 30-32 handle “PING” message to keep the robot process alive. The following if-else structures handle each of the instructions: @repeat, @convert, @ip, and @help.

## Challenge & Solution

- The role of a robot process is confusing to me. At the first beginning, I thought I should have written a server program for a robot, but then figured out that the robot process itself was essentially a client under the Irssi server. Reading through the specifications and slides several times finally cleared the confusion.
- Configuration messages entail a rigorous format. Even missing a colon or introducing one more space character would keep the whole program from working. Thanks to the online documentation and some helpful articles on Stack Overflow (e.g. https://stackoverflow.com/questions/2968408/how-do-i-program-a-simple-irc-bot-in-python (https://stackoverflow.com/questions/2968408/how-do-i-program-a-simple-irc-bot-in-python)), the most tricky part of the format should be covered on these references.
- How to deal with @ip case efficiently may be a good algorithm problem. Thankfully, the input size is limited to be smaller 20, which can be solved with brutal algorithms easily.

## Reflections

The first problem is always to clarify the structure of a problem. For example, the robot itself can be thought of as another user connecting to the Irssi server with direct socket connection. This is quite different from normal users that usually enter a chat room through an application. Understanding the relationship between a robot, users, and Irssi server is the cornerstone for solving the problem.
What follows is socket programming, which requires a background knowledge in the protocols behind the scene. Calling the correct functions is always the easiest part. However, without knowing the mechanism can lead to a error-prone program which is usually hard to debug. A flowchart, like the one listed in the slide, would be helpful to figure out each step of connection and message transmission.

## Screenshot