Homework #2

School ID: Name:

[1] Read Unix Homework Coding Style Guidelines carefully. You must follow these guidelines for your homework. Otherwise, you will get zero points

[2] Test scenario

- **Open the first terminal and run chat Jico to write text messages (Actually, you don't need to create a new user account "Jico". It is enough to use the terminal where you logged in)**
- **Open the second terminal and run chat Izzy to write text messages**
- **Open the third terminal and run chat GD to write text messages**
- **Chat sequences are randomly and one person by one person. We do not consider that three persons chat at the same time. (We do not need threads, synchronization, fork, and etc for this work.)**

[3] The "chat.c" program behaves like the following steps:

- **[Constraints]**
  - **Shared memory key is 20200406already existed, just attach)**
  - **Define the shared memory data structure that supports multi-user chat.**
  - **Each message has a unique ID and so messages will not be duplicated.**
  - **Your messages buffered in the shared memory will also be displayed (echoed…)**
  - **We don't need to scroll up and down text messages**

- **[Procedures]**
  - **Create a shared memory. Attach the shared memory to the chat process (If the shared memory hasalready existed, just attach the shared memory to the chat process)**

- **After you write some messages, you can see display the chat messages in the shared memory (If you want to display new messages arrived at the shared memory, you just write "..". ".." will not be sent to the shared memory**
- **Type "/bye" to quit the chat process.**
- **Run the chatremove (you need to implement your own chatremove.c)**

[3] Your program consists of at least two files and one your own header.

You must build your own Makefile.

==========

1. Put your program source as here (Do not put the screen shot of your source code!! If you insist, you will get zero point)

2. You must show the building result after compiling and linking your source codes. You must show no warnings and errors (Use gcc -Wall option).

(Put a screen shot of your C debugging output)

3. Put a screen shot of output generated by your program. Your output screen shot must be readable for me to verify your chat program.