Instructions. Write an Erlang program that connects **two nodes** and allows the nodes to communicate and send messages to each other (just like in a chat).

Just like in the pingpong example, one node (N_1) will start a process that will wait for a message from someone.

```
(juliet@ics-user)1> chat:init_chat().
```

Another node (N_2) will start a process that will need the long name of the node where to send the message.

```
(romeo@ics-user)1> chat:init_chat2('juliet@ics-user').
```

However, in this exercise, the process N_1 and N_2 should be able to send a message to each other without the other node giving a reply(Meaning replies will come from user input in each node). The chat will be terminated when one node connected sends the string `bye`.

```
(juliet@ics-user)1> chat:init_chat().
Enter Your Name: Juliet
You: O Romeo, Romeo! Wherefore art thou Romeo?
Romeo: Here I am!
You: Deny thy father and refuse thy name
You: Or if thou wilt not, be but sworn my love
You: And I will no longer be a Capulet.
Your partner disconnected
(juliet@ics-user)2>
```

```
(romeo@ics-user)1> chat:init_chat2('juliet@ics-user').
Enter Your Name: Romeo
Juliet: O Romeo, Romeo! Wherefore art thou Romeo?
You: Here I am!
Juliet: Deny thy father and refuse thy name
Juliet: Or if thou wilt not, be but sworn my love
Juliet: And I will no longer be a Capulet.
You: bye
(romeo@ics-user)3>
```

Your program should have the filename: `surname1surname2.erl`.

Grading System

connect two nodes	3 pts
disconnect when the bye message is sent	3 pts
send messages but waits for the reply of the other node	3 pts
send messages and does not wait for the reply of the other node	6 pts

BONUS! (For 4 pts) Three or more persons can talk with each other in a single chat room.