Q1. 當訂閱的 topic 為 /a/test 時，能不能接收到 topic 為 a/test 的訊息? 為什麼?(hint : 比較兩個 topic 的層數)

不能，因為兩個 topic 層數不一樣。MQTT 的 topic 是以階層式設計的，用'/'將各層分開，如果輸入 /a/test 就代表多進入一層空白名稱的根階層底下的 a/test，這樣是三層。但 a/test 只有兩層，因此無法接收 message。如果想要正確接收 message，在 subscriber 端也要輸入對應層數的指令。

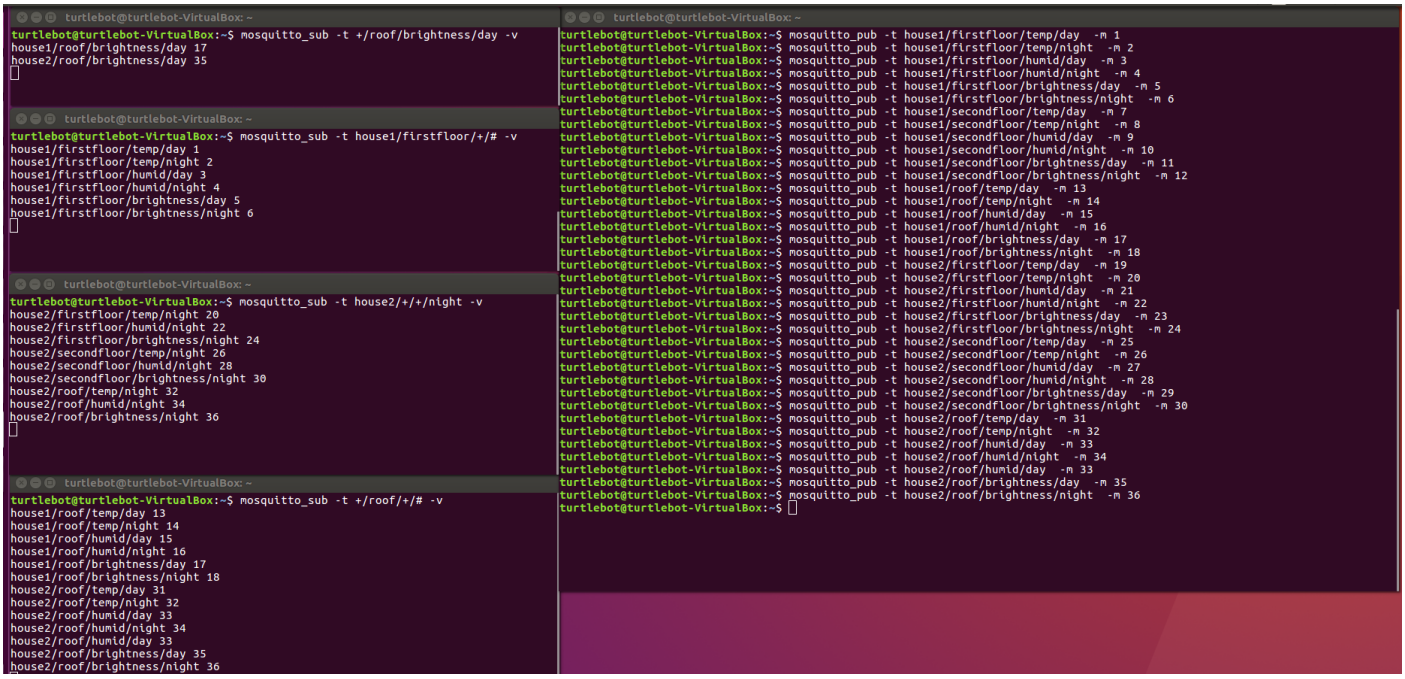
Q2.

Msg format : Place/Floor/Sensor Type/Time/ 共 36 種

- **Place : house1、house2**
- **Floor : firstfloor、secondfloor、roof**
- **Sensor Type : temp、humid、brightness**
- **Time : day、night**

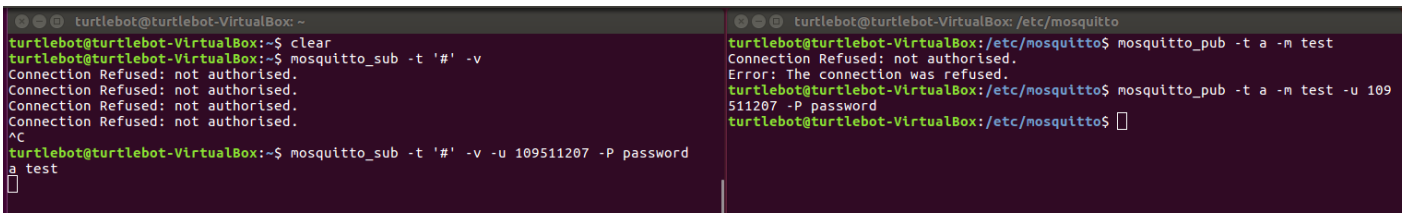
若只想收到以下內容，那麼 subscriber 的 topic 分別要定為什麼？

1. **所有 roof 在 day 的 brightness**
2. **house1 firstfloor 的所有內容**
3. **house2 在 night 的所有內容**
4. **所有 roof 的所有內容**



1. +/roof/brightness/day
2. house1/firstfloor/+/#
3. house2/+/+/night
4. +/roof/+/#

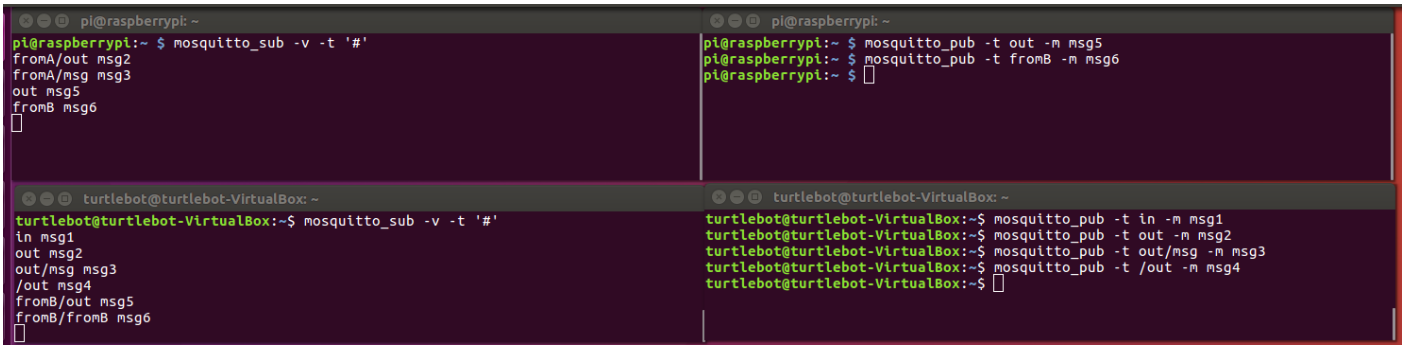
Q3. 排除匿名使用者截圖(參考簡報範例)，帳號設為學號。



新增 allow_anonymous false 來阻擋匿名者操作，只要輸入指令時沒有加上已建立的 user 和 password 的話就會出現 Connection Refused

Q4.

Broker : topic、msg	Broker 1 (A) sub 顯示	Broker 2 (B) sub 顯示
A : in、msg1	in msg1	
A : out、msg2	out msg2	fromA/out msg2
A : out/msg、msg3	out/msg msg3	fromA/msg msg3
A : /out、msg4	/out msg4	
B : out、msg5	fromB/out msg5	out msg5
B : fromB、msg6	fromB/fromB msg6	fromB msg6



Q5. 在 MQTT bridge configuration 中，QoS level 有哪幾種?分別描述不同 level 的功能。

有 0、1 和 2。

0: 最多傳送一次，在網路階層架構中，MQTT 會將產生的訊息交給 TCP 處理。由於 MQTT 程式不清楚最底層的網路是否正常運作，因此不保證訊息會送達。

1: 至少傳送一次，如果接收端收到訊息，就會回傳確認訊息給發送端，若一段時間後，發送端沒有收到確認訊息，就會認定訊息沒有送達，會再重新傳送一次訊息。

2: 確實傳送一次，當接收端收到訊息時會回覆已收到發布訊息，並暫存訊息的封包識別碼，以防重複處理到相同的訊息。

Q6. 在 mosquitto.conf 中，對於設定 cleansession 的說明如下：

`cleansession` [true | false]

> Set the clean session option for this bridge. Setting to *`false`* (the default), means that all subscriptions on the remote broker are kept in case of the network connection dropping. If set to *`true`*, all subscriptions and messages on the remote broker will be cleaned up if the connection drops. Note that setting to *`true`* may cause a large amount of retained messages to be sent each time the bridge reconnects.
>
> If you are using bridges with `cleansession` set to *`false`* (the default), then you may get unexpected behavior from incoming topics if you change what topics you are subscribing to. This is because the remote broker keeps the subscription for the old topic. If you have this problem, connect your bridge with `cleansession` set to *`true`*, then reconnect with cleansession set to *`false`* as normal.

請理解 `cleansession` 參數的目的並回答以下問題：

1. **描述 true | false 分別的作用。**
2. **如果 `cleansession` 設置為 false，而且你更改訂閱的主題，會出現什麼意外的行為？為什麼會發生這種情況？該怎麼解決?**
3. **什麼是"保留消息（retained messages）"，為什麼在 `cleansession` 設置為 true 時可能會出現大量的保留消息被發送的情況？**

1. true 表示在 bridge 斷開 connection 時，broker 將會清除所有訂閱，不保留任何舊的狀態。而 false 表示在 bridge 斷開 connection 時，broker 會保留所有訂閱和未傳送的 message，這樣可以在重新連接繼續處理之前的訂閱和未傳完的 message。
2. broker 會保留之前的訂閱，這樣可能會收到之前取消訂閱的 topic 的 message。會發生這種情況是因為 cleansession 設為 false，broker 保留了先前的訂閱，並在重新 connect 時，繼續使用這些狀態，因此即便更改了訂閱主題，broker 依然記得之前的訂閱。要解決這個問題的方法是在重新連接前將 cleansession 設為 true，這樣 broker 就可以清掉之前的訂閱狀態了，就能夠正確處理新的訂閱 topic，然後在重新連接後，再將 cleansession 設為 false，保留 broker 的訂閱狀態。
3. 保留消息是指 publisher 可以向 broker 發布一條帶有保留標誌的消息，broker 便會保留這條消息，並在有新的 subsrciber 訂閱對應的 topic 時，將其發送給所有訂閱這個 topic 的 subscriber。這使得新的 subscriber 能夠獲取到最近的狀態，而不是等待下一次發布的消息。在 cleansession 設為 true 時，代表 broker 將會清除所有訂閱，因此如果之前的消息是保留消息，那這些消息也會被清除，這樣 broker 就需要重新發送所有的保留消息，確保 subscriber 端能夠獲取最新的狀態，因此如果有多名 subscriber 的話，就需要發送大量的保留消息。

Q7.心得:

MQTT (Message Queuing Telemetry Transport)是為了物聯網而設計的協定，它所需要的頻寬和硬體資源較低，很適合用在低功耗和帶寬有限的 IoT 環境，像是智慧家電或者醫療裝置等等。在 MQTT 中，有向 broker 發布 topic 的 publisher、儲存 publisher 訊息的 broker、從 broker 獲取 topic 的 subscriber，資訊的傳輸是通過 topic 管理的，publisher 有需要分發的資料時，向連接的 broker 發送訊息。broker 會向訂閱此主題的 subscriber 分發訊息。publisher 不需要知道 subscriber 的資料和 IP address，subscriber 也同樣不需要 publisher 的相關資訊。

我自己覺得本次 Lab 並不難，同時扮演 publisher 和 subscriber 的角色也很有趣。