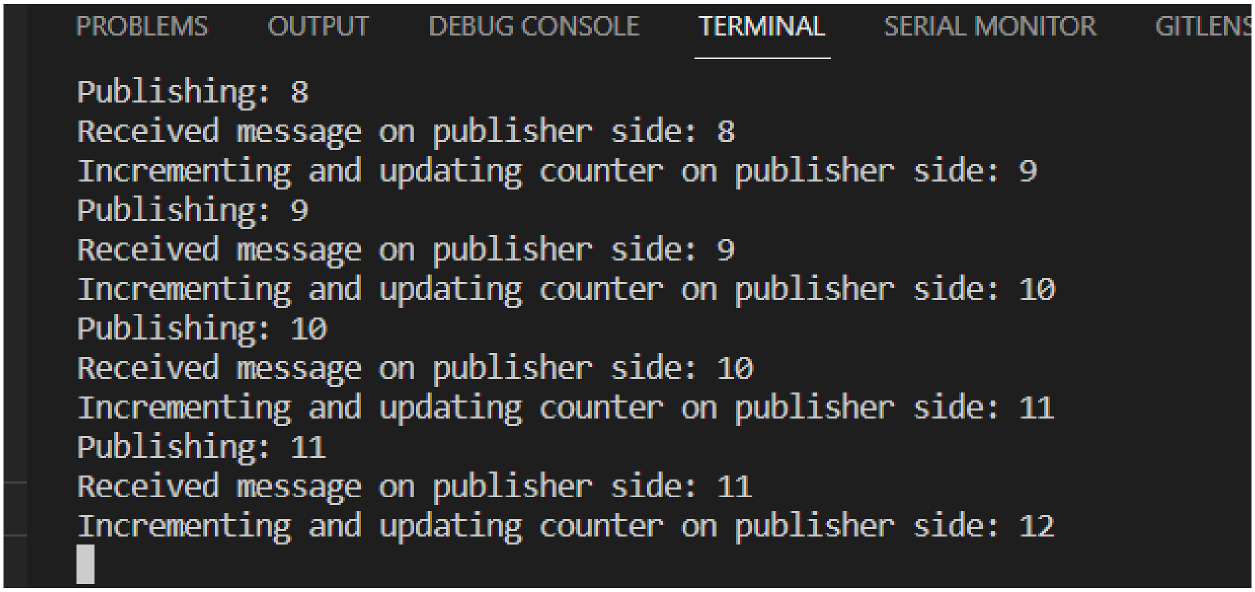
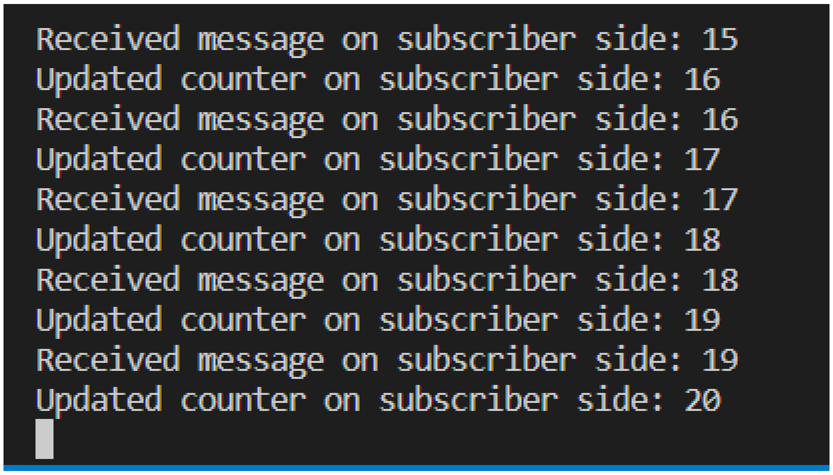
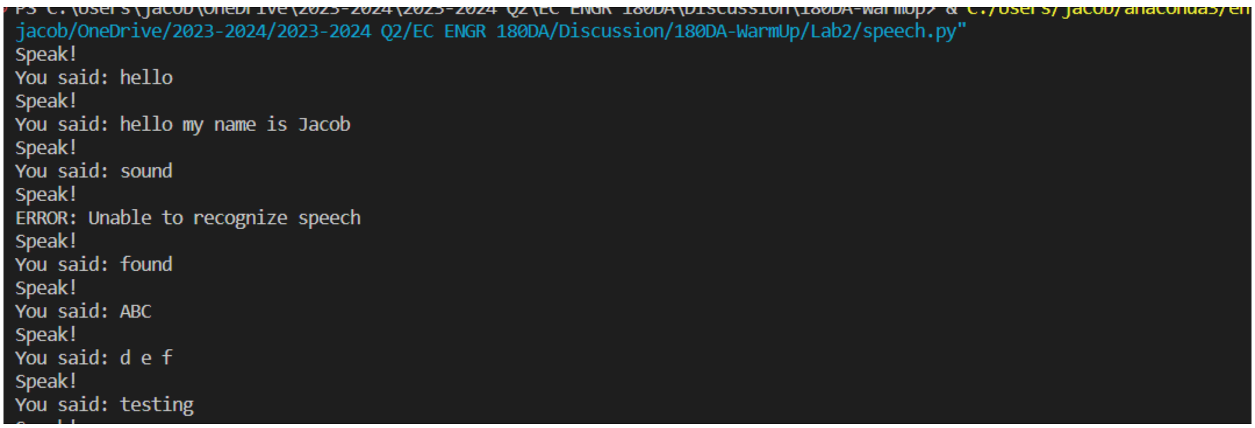


```
Publishing: 8
Received message on publisher side: 8
Incrementing and updating counter on publisher side: 9
Publishing: 9
Received message on publisher side: 9
Incrementing and updating counter on publisher side: 10
Publishing: 10
Received message on publisher side: 10
Incrementing and updating counter on publisher side: 11
Publishing: 11
Received message on publisher side: 11
Incrementing and updating counter on publisher side: 12
```



```
Received message on subscriber side: 15
Updated counter on subscriber side: 16
Received message on subscriber side: 16
Updated counter on subscriber side: 17
Received message on subscriber side: 17
Updated counter on subscriber side: 18
Received message on subscriber side: 18
Updated counter on subscriber side: 19
Received message on subscriber side: 19
Updated counter on subscriber side: 20
```



```
PS C:\Users\Jacob\OneDrive\2023-2024\2023-2024 Q2\EC ENGR 180DA\Discussion\180DA-WarmUp> & C:/Users/Jacob/anaconda3/en
jacob/OneDrive/2023-2024/2023-2024 Q2/EC ENGR 180DA/Discussion/180DA-WarmUp/Lab2/speech.py"
Speak!
You said: hello
Speak!
You said: hello my name is Jacob
Speak!
You said: sound
Speak!
ERROR: Unable to recognize speech
Speak!
You said: found
Speak!
You said: ABC
Speak!
You said: d e f
Speak!
You said: testing
```

1. MQTT is very good for providing updates on the most recent value of a sensor or something similar. Its latency is relatively low and it is easy to use, but the fact that only string data can be sent is a little bit limiting. I think I will use it in the project to send messages from sensors to a central computer, which will run the

actual heavy program. For example, it could be a button sending “pressed” only when it is pressed down, but I wouldn’t want to continuously send the state of the button.

2. The longer phrases seemed to work better, the single letters were extremely hard to recognize. Sound and found were usually distinguishable, but the letters were very hard. I did notice that the louder the lab was, the more likely the recognizer was to fail.

- For my project, I plan to use speech to activate some powerups in the game, so that the users can activate them hands free.
- I think simple phrases or words will likely be used, like “freeze” or “extra life”, because they are easy to remember for the user and easy to recognize as they are full words.
- I need full accuracy, so I am considering obtaining a better microphone for my laptop to be able to more easily hear the players. If a user missed their powerup, it would definitely be a hindrance to the experience and feel unfair.
- I would need the room to be relatively quiet for this to work, at least a low noise floor. It is working ok in the lab but that is with me right up to the microphone.