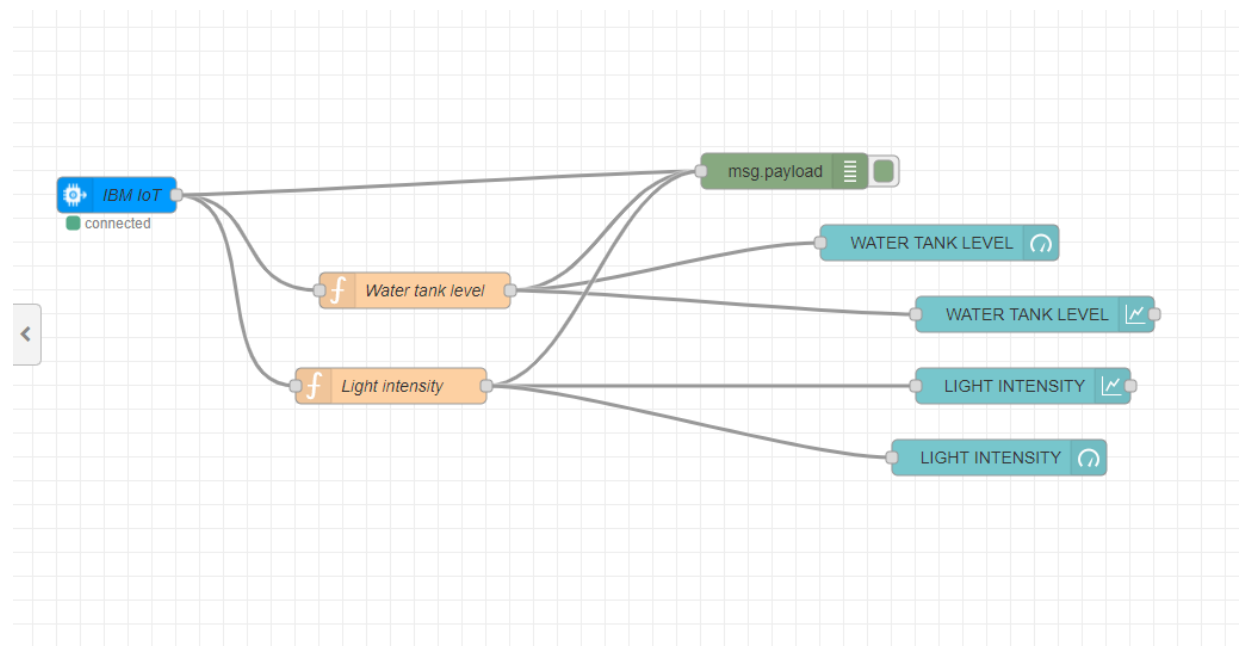


# ASSIGNMENT 3

Develop a code to upload the water tank level and light intensity values to the IBM IoT platform and visualize them in the web application.

## NODE RED

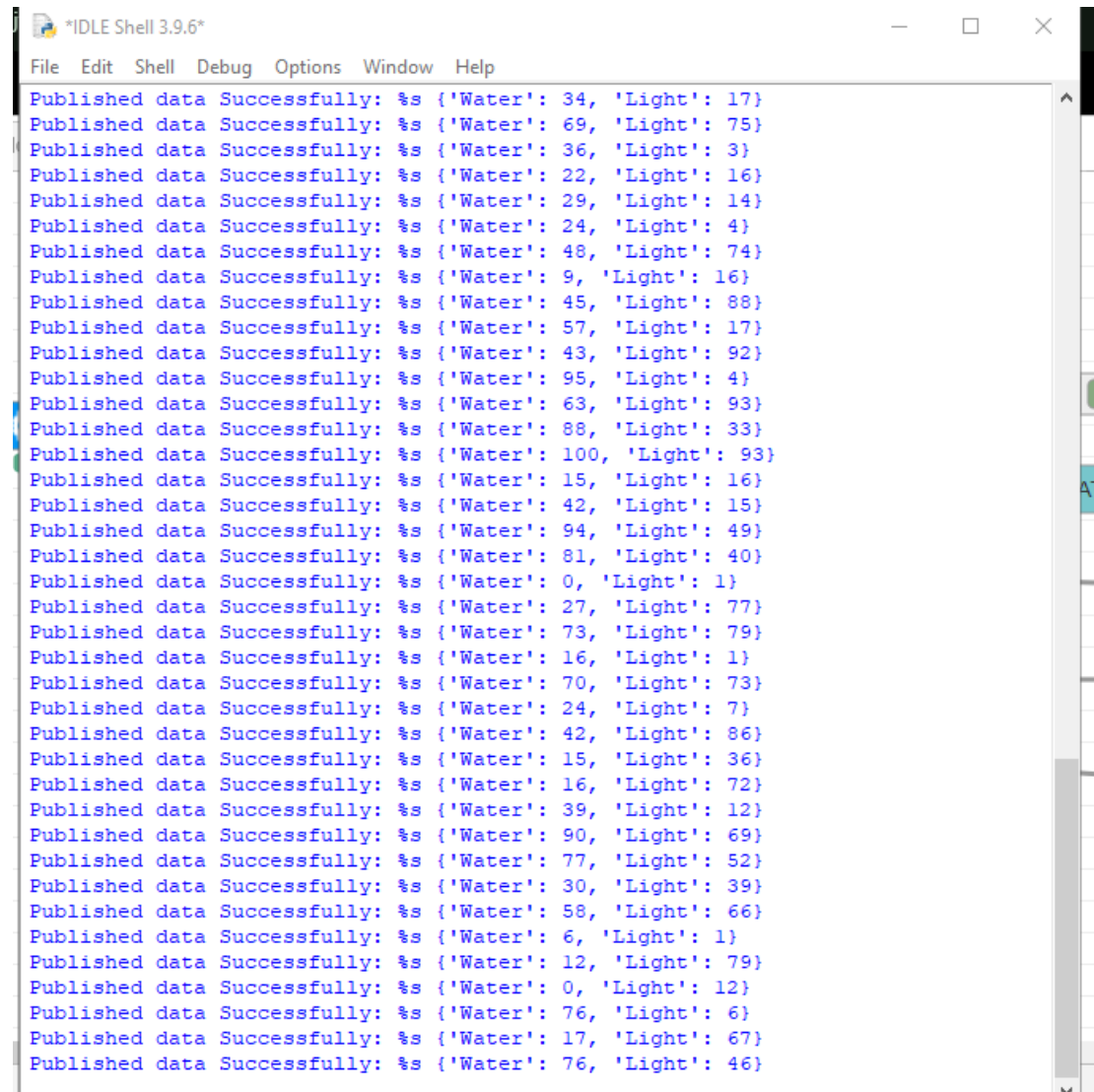


# DEBUG

The screenshot shows the Node-RED interface with a flow on the left and the debug console on the right. The flow starts with a `msg.payload` node, which branches into four output nodes: `WATER TANK L`, `WA`, `LIG`, and `LIGHT`. The debug console is titled "debug" and shows a list of messages. The messages are as follows:

Timestamp	Node ID	msg.payload
7/16/2021, 3:01:38 PM	938e9c91.7b3ad	<code>{ Water: 87, Light: 67 }</code>
7/16/2021, 3:01:39 PM	938e9c91.7b3ad	87
7/16/2021, 3:01:39 PM	938e9c91.7b3ad	67
7/16/2021, 3:01:41 PM	938e9c91.7b3ad	<code>{ Water: 93, Light: 32 }</code>
7/16/2021, 3:01:41 PM	938e9c91.7b3ad	93
7/16/2021, 3:01:41 PM	938e9c91.7b3ad	32

# PYTHON SHELL



The screenshot shows a Python IDLE Shell window titled '\*IDLE Shell 3.9.6\*'. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area displays 40 lines of output, each starting with 'Published data Successfully: %s' followed by a dictionary containing 'Water' and 'Light' values. The values for 'Water' range from 0 to 100, and 'Light' values range from 1 to 93. The window includes standard OS window controls (minimize, maximize, close) and a vertical scrollbar on the right.

```
Published data Successfully: %s {'Water': 34, 'Light': 17}
Published data Successfully: %s {'Water': 69, 'Light': 75}
Published data Successfully: %s {'Water': 36, 'Light': 3}
Published data Successfully: %s {'Water': 22, 'Light': 16}
Published data Successfully: %s {'Water': 29, 'Light': 14}
Published data Successfully: %s {'Water': 24, 'Light': 4}
Published data Successfully: %s {'Water': 48, 'Light': 74}
Published data Successfully: %s {'Water': 9, 'Light': 16}
Published data Successfully: %s {'Water': 45, 'Light': 88}
Published data Successfully: %s {'Water': 57, 'Light': 17}
Published data Successfully: %s {'Water': 43, 'Light': 92}
Published data Successfully: %s {'Water': 95, 'Light': 4}
Published data Successfully: %s {'Water': 63, 'Light': 93}
Published data Successfully: %s {'Water': 88, 'Light': 33}
Published data Successfully: %s {'Water': 100, 'Light': 93}
Published data Successfully: %s {'Water': 15, 'Light': 16}
Published data Successfully: %s {'Water': 42, 'Light': 15}
Published data Successfully: %s {'Water': 94, 'Light': 49}
Published data Successfully: %s {'Water': 81, 'Light': 40}
Published data Successfully: %s {'Water': 0, 'Light': 1}
Published data Successfully: %s {'Water': 27, 'Light': 77}
Published data Successfully: %s {'Water': 73, 'Light': 79}
Published data Successfully: %s {'Water': 16, 'Light': 1}
Published data Successfully: %s {'Water': 70, 'Light': 73}
Published data Successfully: %s {'Water': 24, 'Light': 7}
Published data Successfully: %s {'Water': 42, 'Light': 86}
Published data Successfully: %s {'Water': 15, 'Light': 36}
Published data Successfully: %s {'Water': 16, 'Light': 72}
Published data Successfully: %s {'Water': 39, 'Light': 12}
Published data Successfully: %s {'Water': 90, 'Light': 69}
Published data Successfully: %s {'Water': 77, 'Light': 52}
Published data Successfully: %s {'Water': 30, 'Light': 39}
Published data Successfully: %s {'Water': 58, 'Light': 66}
Published data Successfully: %s {'Water': 6, 'Light': 1}
Published data Successfully: %s {'Water': 12, 'Light': 79}
Published data Successfully: %s {'Water': 0, 'Light': 12}
Published data Successfully: %s {'Water': 76, 'Light': 6}
Published data Successfully: %s {'Water': 17, 'Light': 67}
Published data Successfully: %s {'Water': 76, 'Light': 46}
```

# OUTPUT CHART AND GAUGE

