22AIE442

Labsheet 3

Name : Anuvind M P

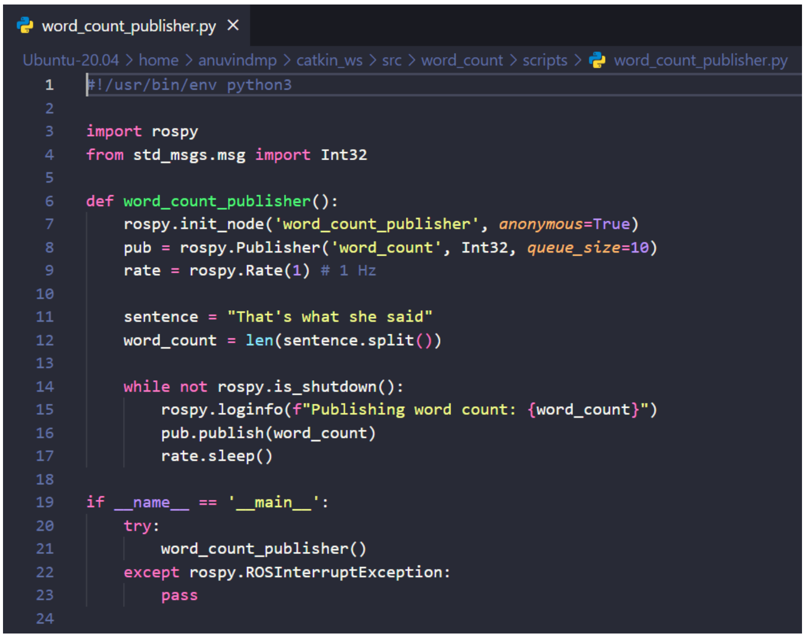
Roll no : AM.EN.U4AIE22010

Q1)

a) Write a program in python to count the number of words in a string.
b) Embed the pseudocode in the publisher program.
c) Write a subscriber to read the result

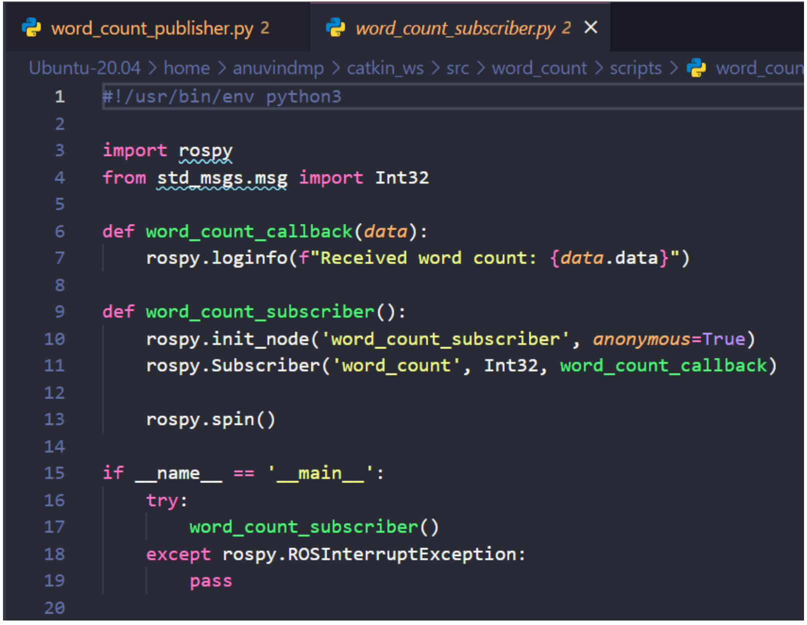
I) Without user input from console

Code (Publisher):


word_count_publisher.py ×
Ubuntu-20.04 > home > anuvindmp > catkin_ws > src > word_count > scripts > word_count_publisher.py

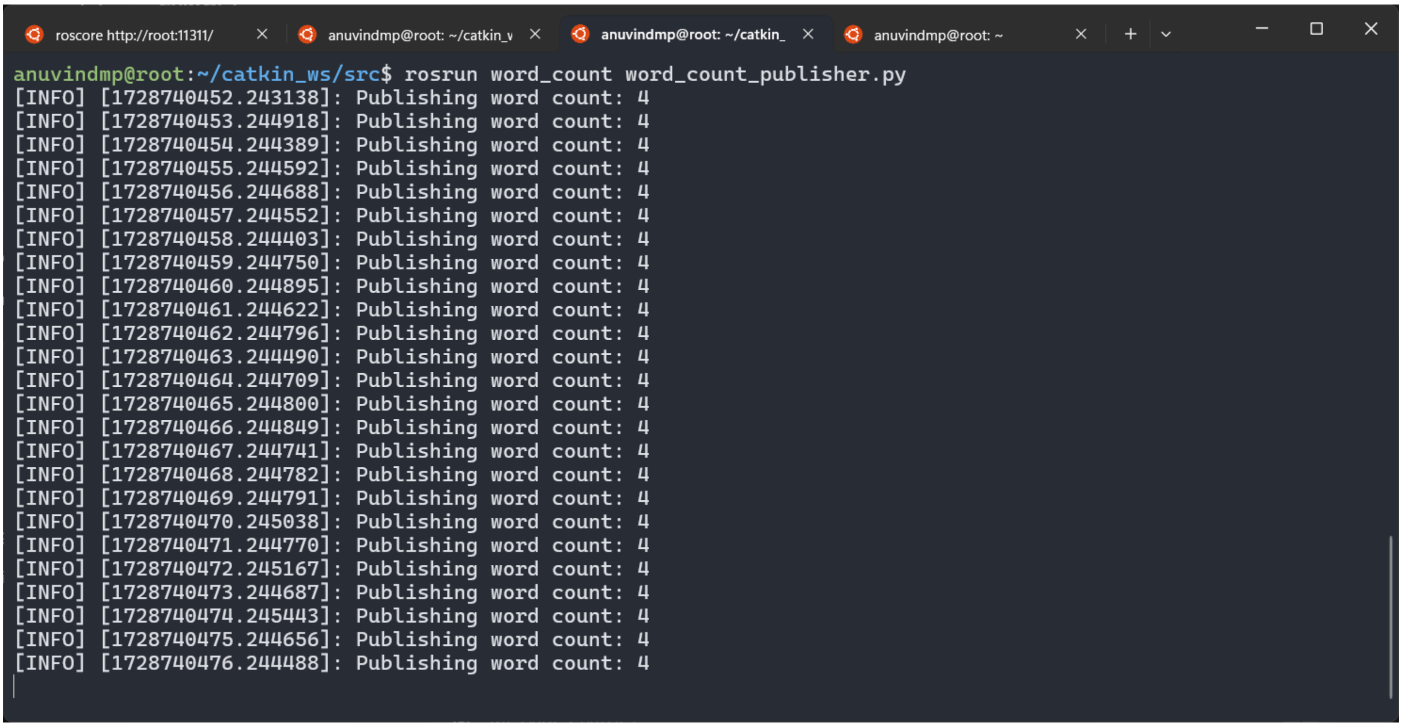
```python
#!/usr/bin/env python3

import rospy
from std_msgs.msg import Int32

def word_count_publisher():
    rospy.init_node('word_count_publisher', anonymous=True)
    pub = rospy.Publisher('word_count', Int32, queue_size=10)
    rate = rospy.Rate(1) # 1 Hz

    sentence = "That's what she said"
    word_count = len(sentence.split())

    while not rospy.is_shutdown():
        rospy.loginfo(f"Publishing word count: {word_count}")
        pub.publish(word_count)
        rate.sleep()

if __name__ == '__main__':
    try:
        word_count_publisher()
    except rospy.ROSInterruptException:
        pass
```

Code (Subscriber):


word_count_publisher.py 2
word_count_subscriber.py 2 ×
Ubuntu-20.04 > home > anuvindmp > catkin_ws > src > word_count > scripts > word_coun

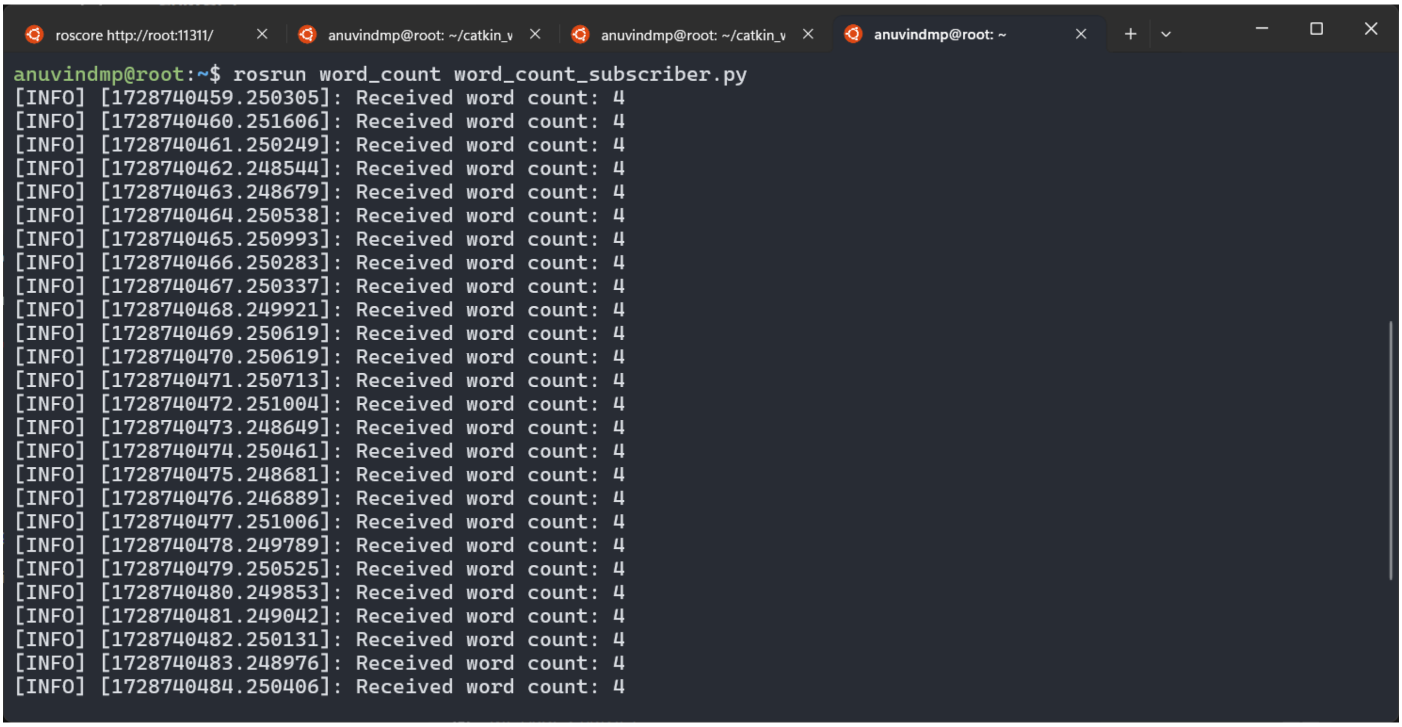
```python
#!/usr/bin/env python3

import rospy
from std_msgs.msg import Int32

def word_count_callback(data):
    rospy.loginfo(f"Received word count: {data.data}")

def word_count_subscriber():
    rospy.init_node('word_count_subscriber', anonymous=True)
    rospy.Subscriber('word_count', Int32, word_count_callback)

    rospy.spin()

if __name__ == '__main__':
    try:
        word_count_subscriber()
    except rospy.ROSInterruptException:
        pass
```

Output (Publisher):


roscore http://root:11311/
anuvindmp@root: ~/catkin_w
anuvindmp@root: ~/catkin_
anuvindmp@root: ~

```
anuvindmp@root:~/catkin_ws/src$ rosrun word_count word_count_publisher.py
[INFO] [1728740452.243138]: Publishing word count: 4
[INFO] [1728740453.244918]: Publishing word count: 4
[INFO] [1728740454.244389]: Publishing word count: 4
[INFO] [1728740455.244592]: Publishing word count: 4
[INFO] [1728740456.244688]: Publishing word count: 4
[INFO] [1728740457.244552]: Publishing word count: 4
[INFO] [1728740458.244403]: Publishing word count: 4
[INFO] [1728740459.244750]: Publishing word count: 4
[INFO] [1728740460.244895]: Publishing word count: 4
[INFO] [1728740461.244622]: Publishing word count: 4
[INFO] [1728740462.244796]: Publishing word count: 4
[INFO] [1728740463.244490]: Publishing word count: 4
[INFO] [1728740464.244709]: Publishing word count: 4
[INFO] [1728740465.244800]: Publishing word count: 4
[INFO] [1728740466.244849]: Publishing word count: 4
[INFO] [1728740467.244741]: Publishing word count: 4
[INFO] [1728740468.244782]: Publishing word count: 4
[INFO] [1728740469.244791]: Publishing word count: 4
[INFO] [1728740470.245038]: Publishing word count: 4
[INFO] [1728740471.244770]: Publishing word count: 4
[INFO] [1728740472.245167]: Publishing word count: 4
[INFO] [1728740473.244687]: Publishing word count: 4
[INFO] [1728740474.245443]: Publishing word count: 4
[INFO] [1728740475.244656]: Publishing word count: 4
[INFO] [1728740476.244488]: Publishing word count: 4
```

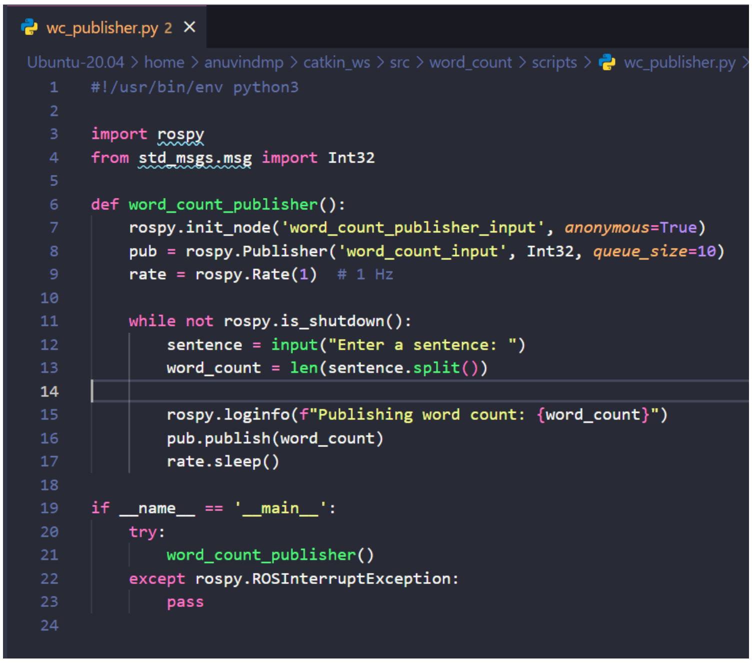
Output (Subscriber):


roscore http://root:11311/
anuvindmp@root: ~/catkin_w
anuvindmp@root: ~/catkin_w
anuvindmp@root: ~

```
anuvindmp@root:~$ rosrun word_count word_count_subscriber.py
[INFO] [1728740459.250305]: Received word count: 4
[INFO] [1728740460.251606]: Received word count: 4
[INFO] [1728740461.250249]: Received word count: 4
[INFO] [1728740462.248544]: Received word count: 4
[INFO] [1728740463.248679]: Received word count: 4
[INFO] [1728740464.250538]: Received word count: 4
[INFO] [1728740465.250993]: Received word count: 4
[INFO] [1728740466.250283]: Received word count: 4
[INFO] [1728740467.250337]: Received word count: 4
[INFO] [1728740468.249921]: Received word count: 4
[INFO] [1728740469.250619]: Received word count: 4
[INFO] [1728740470.250619]: Received word count: 4
[INFO] [1728740471.250713]: Received word count: 4
[INFO] [1728740472.251004]: Received word count: 4
[INFO] [1728740473.248649]: Received word count: 4
[INFO] [1728740474.250461]: Received word count: 4
[INFO] [1728740475.248681]: Received word count: 4
[INFO] [1728740476.246889]: Received word count: 4
[INFO] [1728740477.251006]: Received word count: 4
[INFO] [1728740478.249789]: Received word count: 4
[INFO] [1728740479.250525]: Received word count: 4
[INFO] [1728740480.249853]: Received word count: 4
[INFO] [1728740481.249042]: Received word count: 4
[INFO] [1728740482.250131]: Received word count: 4
[INFO] [1728740483.248976]: Received word count: 4
[INFO] [1728740484.250406]: Received word count: 4
```

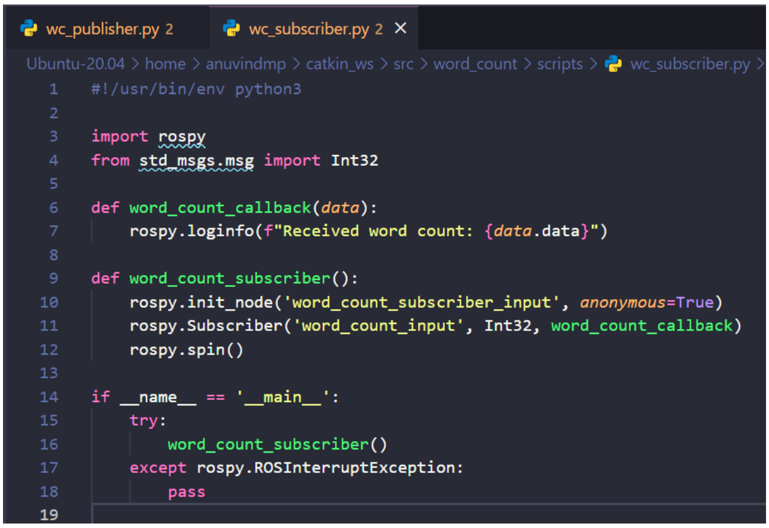
II) With user input from console

Code (Publisher):


wc_publisher.py 2 ×
Ubuntu-20.04 > home > anuvindmp > catkin_ws > src > word_count > scripts > wc_publisher.py >

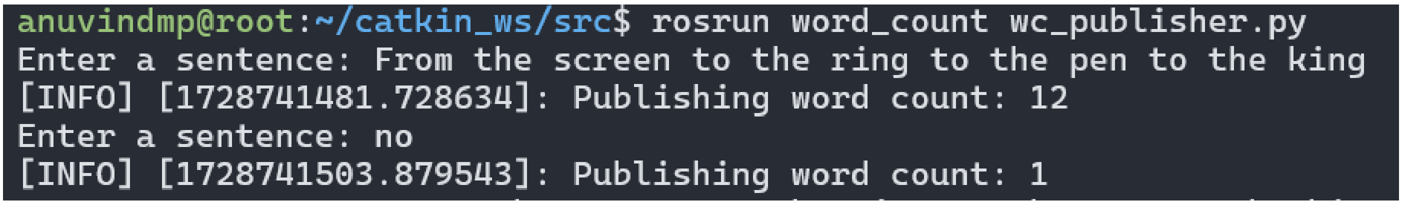
```python
#!/usr/bin/env python3

import rospy
from std_msgs.msg import Int32

def word_count_publisher():
    rospy.init_node('word_count_publisher_input', anonymous=True)
    pub = rospy.Publisher('word_count_input', Int32, queue_size=10)
    rate = rospy.Rate(1)  # 1 Hz

    while not rospy.is_shutdown():
        sentence = input("Enter a sentence: ")
        word_count = len(sentence.split())

        rospy.loginfo(f"Publishing word count: {word_count}")
        pub.publish(word_count)
        rate.sleep()

if __name__ == '__main__':
    try:
        word_count_publisher()
    except rospy.ROSInterruptException:
        pass

```

Code (Subscriber):


wc_publisher.py 2
wc_subscriber.py 2 ×
Ubuntu-20.04 > home > anuvindmp > catkin_ws > src > word_count > scripts > wc_subscriber.py >

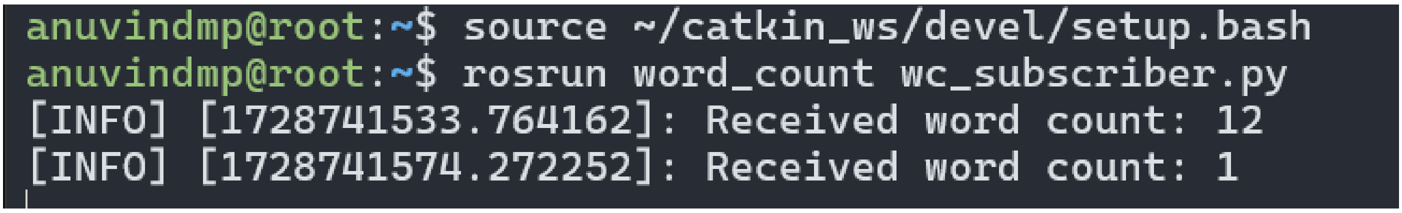
```python
#!/usr/bin/env python3

import rospy
from std_msgs.msg import Int32

def word_count_callback(data):
    rospy.loginfo(f"Received word count: {data.data}")

def word_count_subscriber():
    rospy.init_node('word_count_subscriber_input', anonymous=True)
    rospy.Subscriber('word_count_input', Int32, word_count_callback)
    rospy.spin()

if __name__ == '__main__':
    try:
        word_count_subscriber()
    except rospy.ROSInterruptException:
        pass

```

Output (Publisher):



```
anuvindmp@root:~/catkin_ws/src$ rosrun word_count wc_publisher.py
Enter a sentence: From the screen to the ring to the pen to the king
[INFO] [1728741481.728634]: Publishing word count: 12
Enter a sentence: no
[INFO] [1728741503.879543]: Publishing word count: 1
```

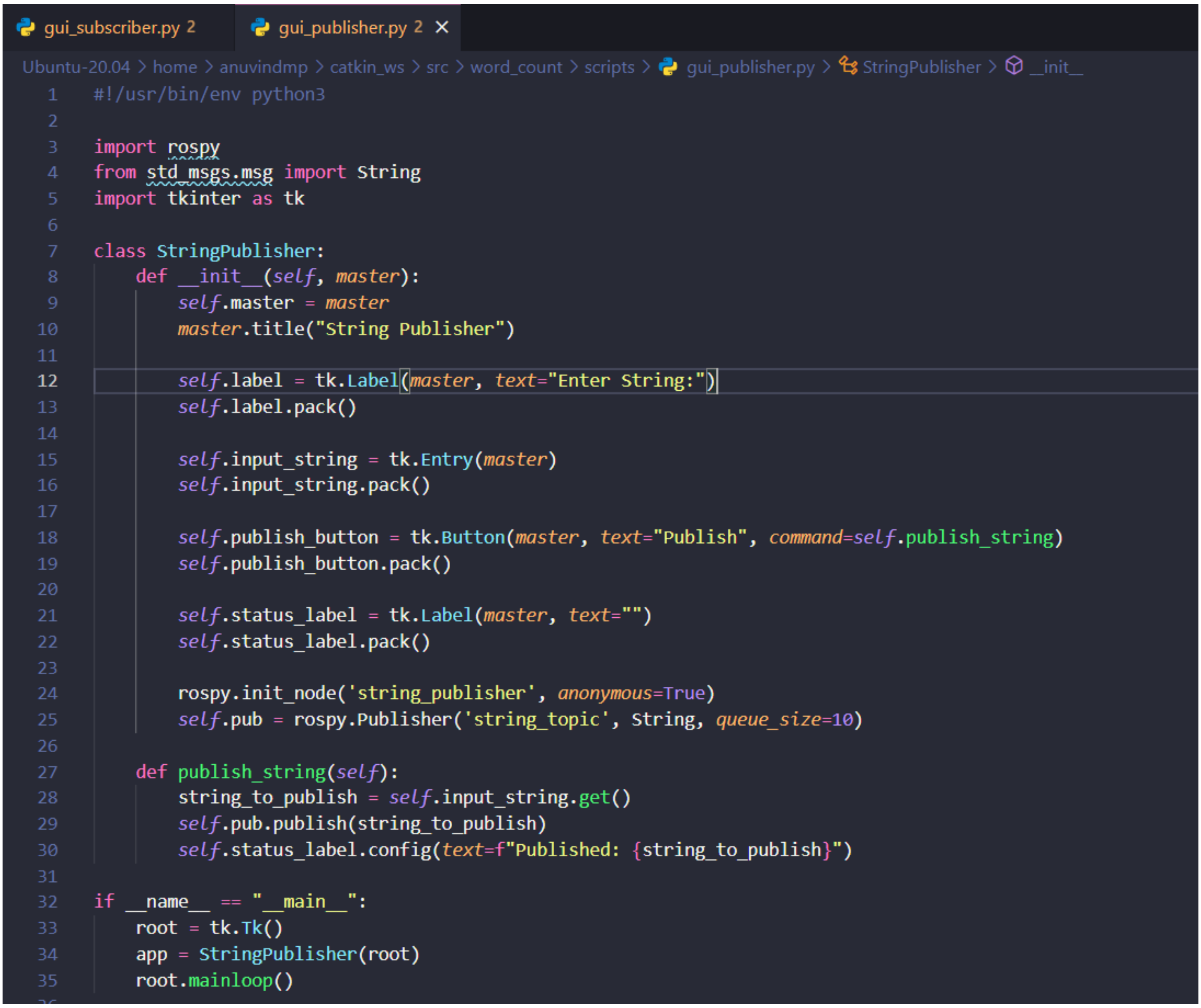
Output (Subscriber):



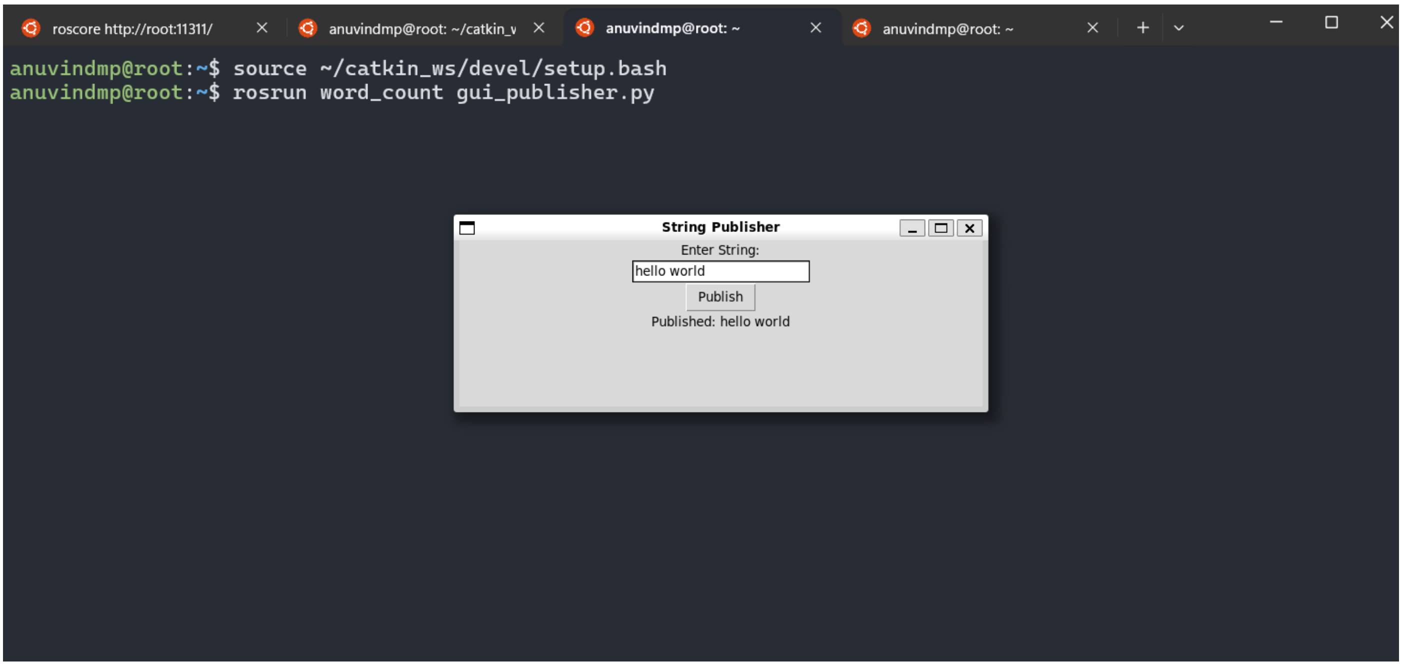
```
anuvindmp@root:~$ source ~/catkin_ws/devel/setup.bash
anuvindmp@root:~$ rosrun word_count wc_subscriber.py
[INFO] [1728741533.764162]: Received word count: 12
[INFO] [1728741574.272252]: Received word count: 1
```

Q2)

a) Create a GUI to publish data


gui_subscriber.py 2
gui_publisher.py 2
Ubuntu-20.04 > home > anuvindmp > catkin_ws > src > word_count > scripts > gui_publisher.py > StringPublisher > __init__

```python
#!/usr/bin/env python3

import rospy
from std_msgs.msg import String
import tkinter as tk

class StringPublisher:
    def __init__(self, master):
        self.master = master
        master.title("String Publisher")

        self.label = tk.Label(master, text="Enter String:")
        self.label.pack()

        self.input_string = tk.Entry(master)
        self.input_string.pack()

        self.publish_button = tk.Button(master, text="Publish", command=self.publish_string)
        self.publish_button.pack()

        self.status_label = tk.Label(master, text="")
        self.status_label.pack()

        rospy.init_node('string_publisher', anonymous=True)
        self.pub = rospy.Publisher('string_topic', String, queue_size=10)

    def publish_string(self):
        string_to_publish = self.input_string.get()
        self.pub.publish(string_to_publish)
        self.status_label.config(text=f"Published: {string_to_publish}")

if __name__ == "__main__":
    root = tk.Tk()
    app = StringPublisher(root)
    root.mainloop()
```

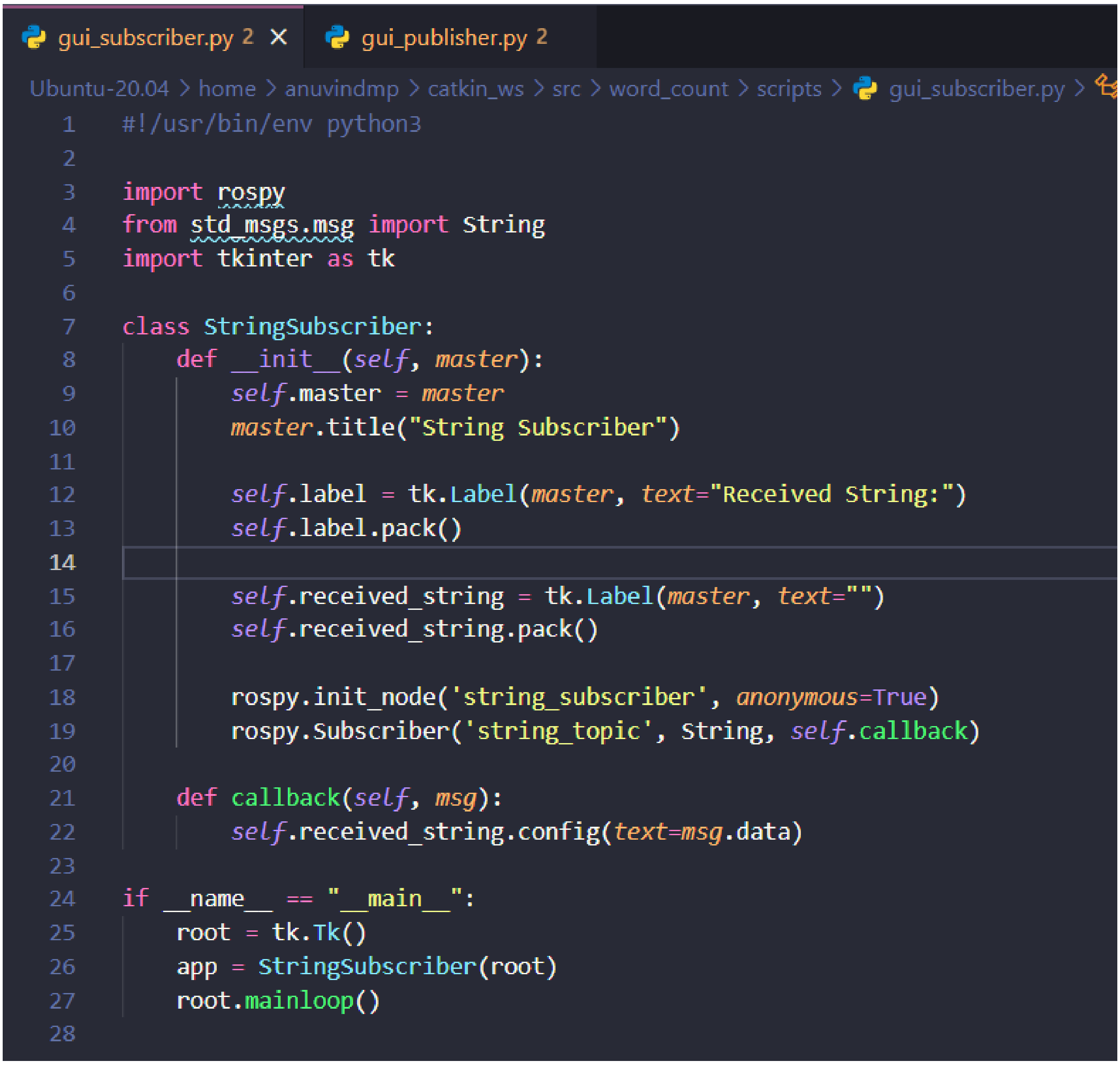


```
anuvindmp@root:~$ source ~/catkin_ws/devel/setup.bash
anuvindmp@root:~$ rosrun word_count gui_publisher.py
```

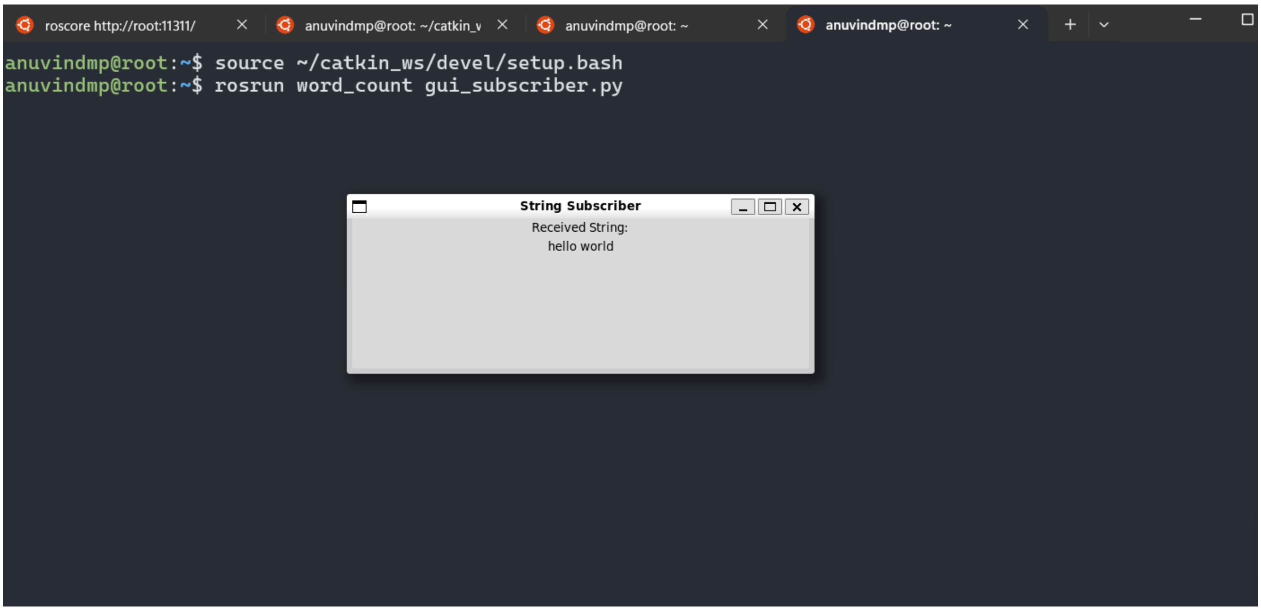
Enter String: hello world

Published: hello world

b) Create a GUI to subscribe data


gui_subscriber.py 2
gui_publisher.py 2
Ubuntu-20.04 > home > anuvindmp > catkin_ws > src > word_count > scripts > gui_subscriber.py >

```python
#!/usr/bin/env python3

import rospy
from std_msgs.msg import String
import tkinter as tk

class StringSubscriber:
    def __init__(self, master):
        self.master = master
        master.title("String Subscriber")

        self.label = tk.Label(master, text="Received String:")
        self.label.pack()

        self.received_string = tk.Label(master, text="")
        self.received_string.pack()

        rospy.init_node('string_subscriber', anonymous=True)
        rospy.Subscriber('string_topic', String, self.callback)

    def callback(self, msg):
        self.received_string.config(text=msg.data)

if __name__ == "__main__":
    root = tk.Tk()
    app = StringSubscriber(root)
    root.mainloop()

```



```
anuvindmp@root:~$ source ~/catkin_ws/devel/setup.bash
anuvindmp@root:~$ rosrun word_count gui_subscriber.py
```

String Subscriber

Received String:

hello world