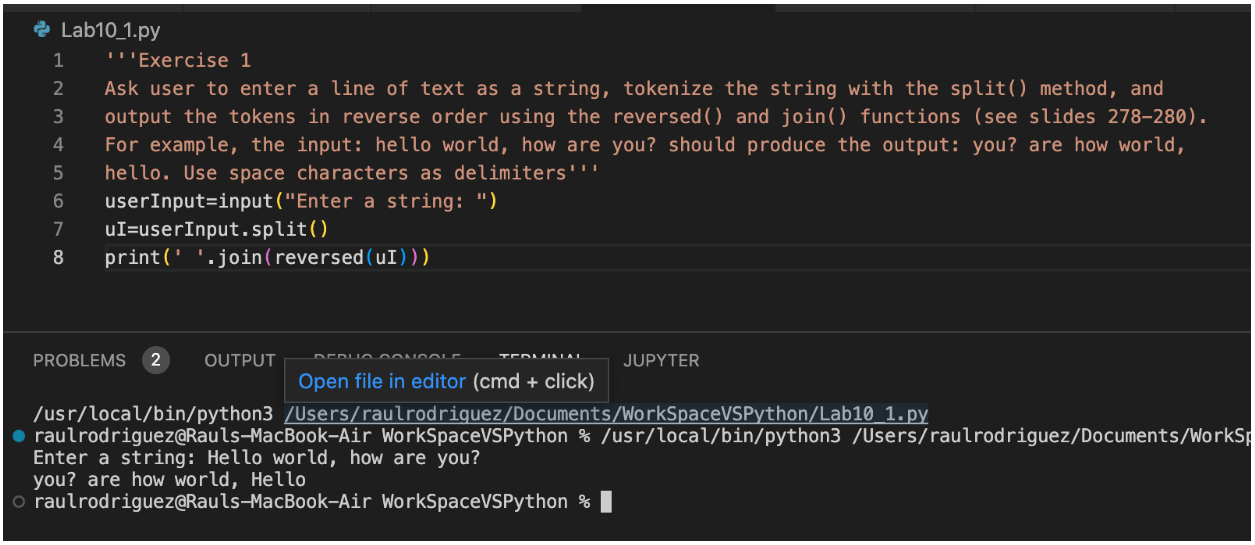
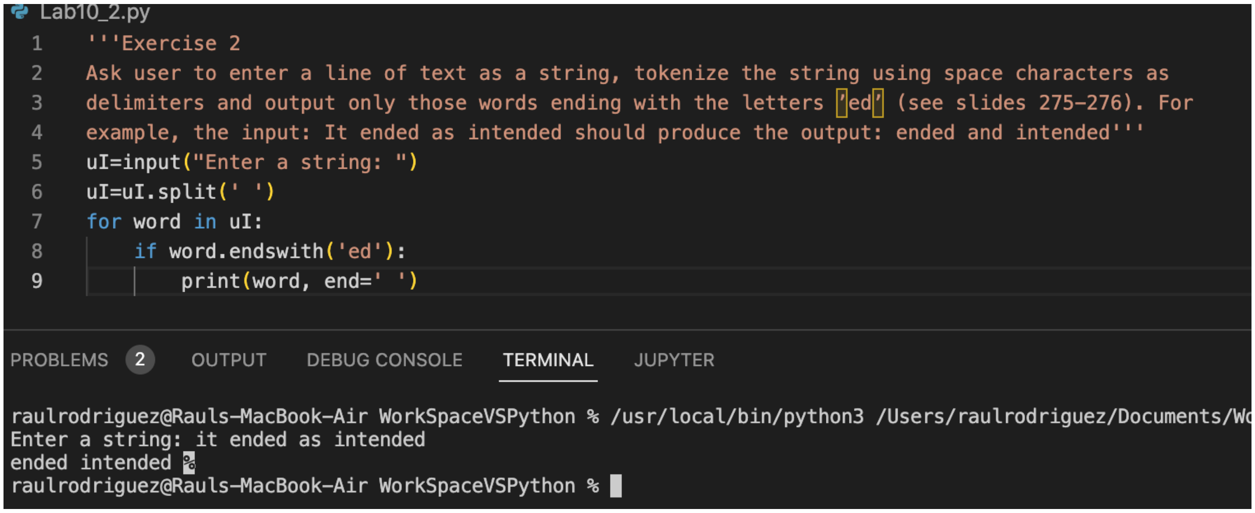
Raul Rodriguez



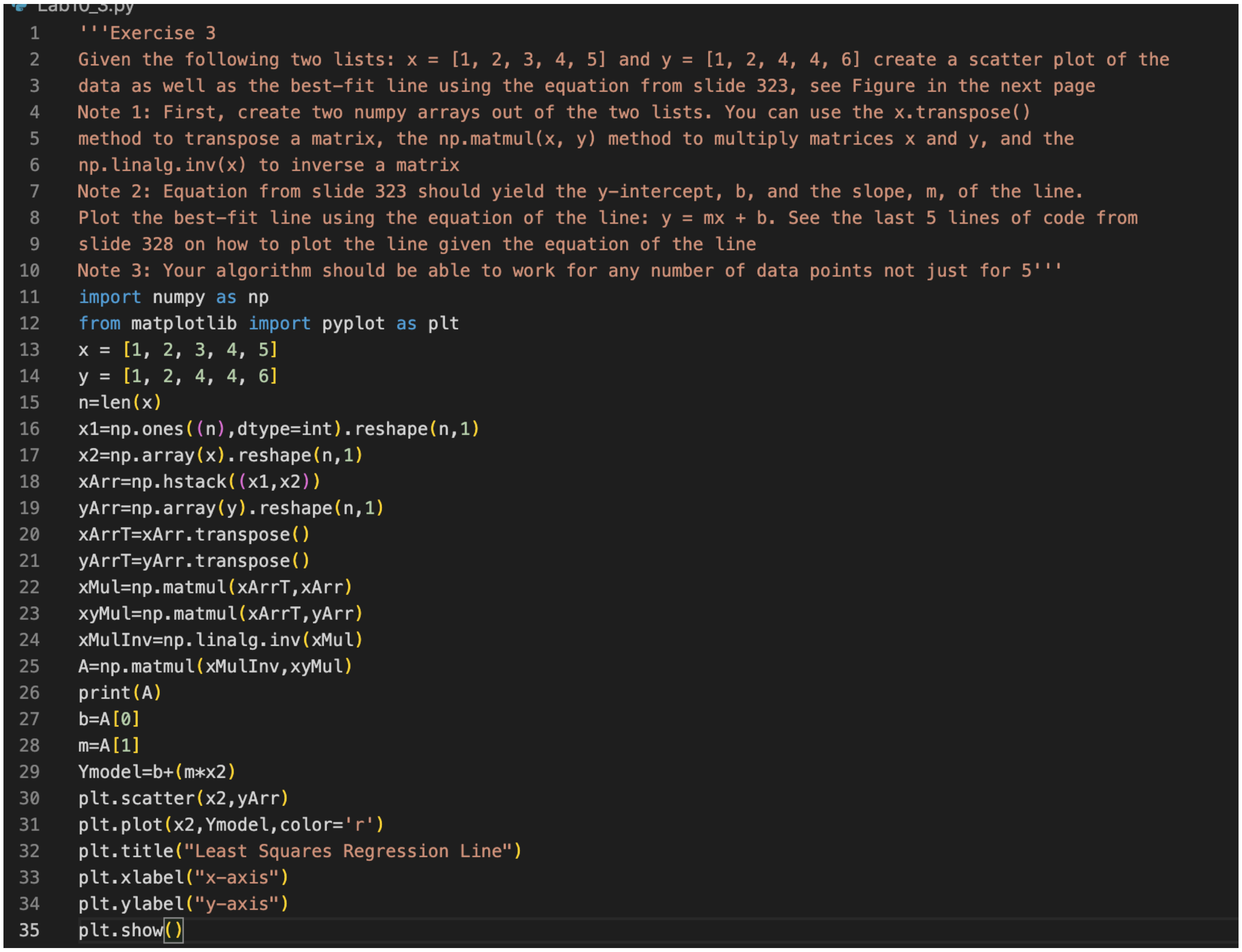
```python
'''Exercise 1
Ask user to enter a line of text as a string, tokenize the string with the split() method, and
output the tokens in reverse order using the reversed() and join() functions (see slides 278-280).
For example, the input: hello world, how are you? should produce the output: you? are how world,
hello. Use space characters as delimiters'''
userInput=input("Enter a string: ")
uI=userInput.split()
print(' '.join(reversed(uI)))
```

```
/usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/Lab10_1.py
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSp
Enter a string: Hello world, how are you?
you? are how world, Hello
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % 
```



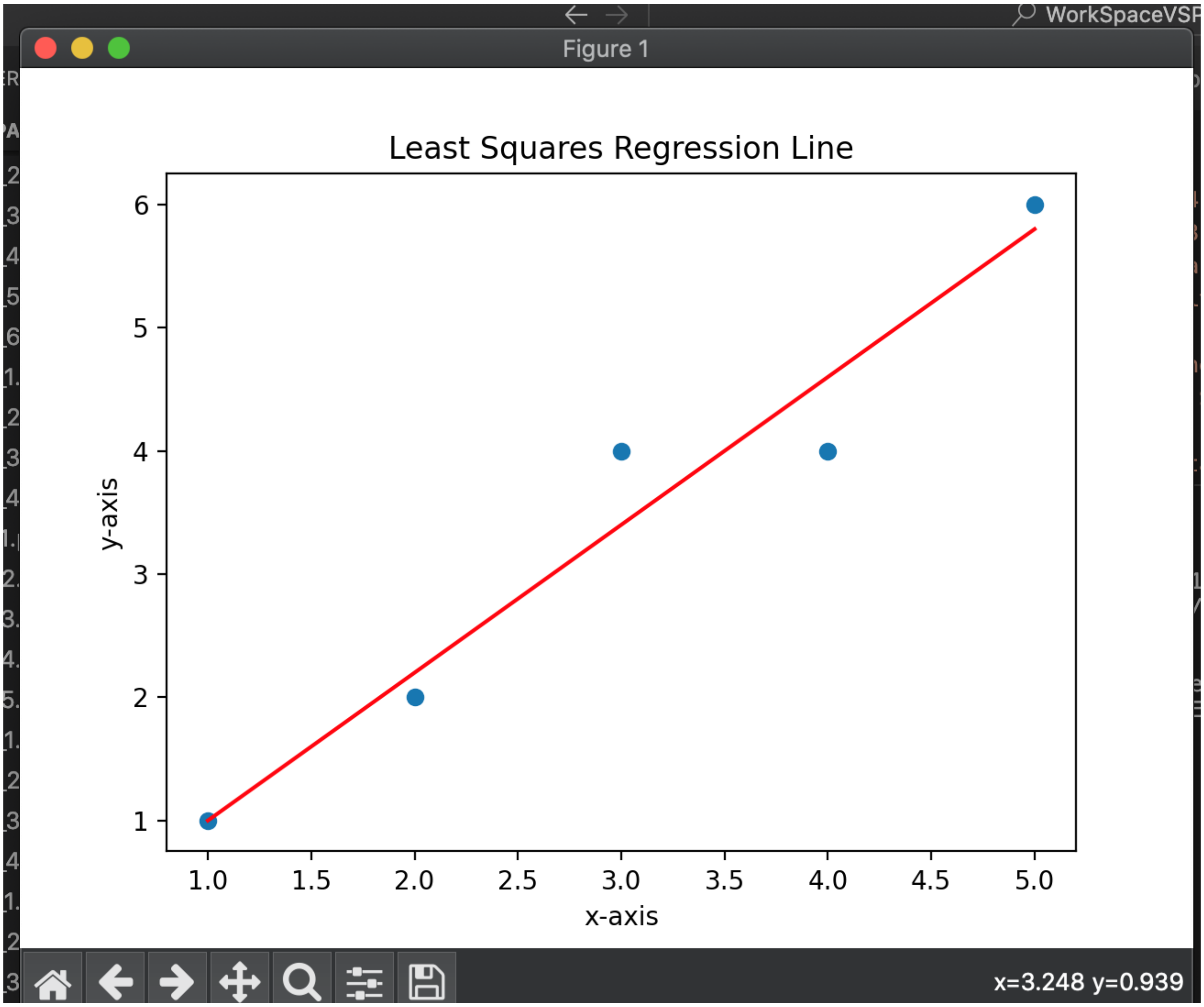
```python
'''Exercise 2
Ask user to enter a line of text as a string, tokenize the string using space characters as
delimiters and output only those words ending with the letters ’ed’ (see slides 275-276). For
example, the input: It ended as intended should produce the output: ended and intended'''
uI=input("Enter a string: ")
uI=uI.split(' ')
for word in uI:
    if word.endswith('ed'):
        print(word, end=' ')
```

```
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/Wo
Enter a string: it ended as intended
ended intended %
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % 
```



Lab10_3.py

```python
'''Exercise 3
Given the following two lists: x = [1, 2, 3, 4, 5] and y = [1, 2, 4, 4, 6] create a scatter plot of the
data as well as the best-fit line using the equation from slide 323, see Figure in the next page
Note 1: First, create two numpy arrays out of the two lists. You can use the x.transpose()
method to transpose a matrix, the np.matmul(x, y) method to multiply matrices x and y, and the
np.linalg.inv(x) to inverse a matrix
Note 2: Equation from slide 323 should yield the y-intercept, b, and the slope, m, of the line.
Plot the best-fit line using the equation of the line: y = mx + b. See the last 5 lines of code from
slide 328 on how to plot the line given the equation of the line
Note 3: Your algorithm should be able to work for any number of data points not just for 5'''
import numpy as np
from matplotlib import pyplot as plt
x = [1, 2, 3, 4, 5]
y = [1, 2, 4, 4, 6]
n=len(x)
x1=np.ones((n),dtype=int).reshape(n,1)
x2=np.array(x).reshape(n,1)
xArr=np.hstack((x1,x2))
yArr=np.array(y).reshape(n,1)
xArrT=xArr.transpose()
yArrT=yArr.transpose()
xMul=np.matmul(xArrT,xArr)
xyMul=np.matmul(xArrT,yArr)
xMulInv=np.linalg.inv(xMul)
A=np.matmul(xMulInv,xyMul)
print(A)
b=A[0]
m=A[1]
Ymodel=b+(m*x2)
plt.scatter(x2,yArr)
plt.plot(x2,Ymodel,color='r')
plt.title("Least Squares Regression Line")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
```


WorkSpaceVSP
Figure 1
Least Squares Regression Line
y-axis
1
2
3
4
5
6
x-axis
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
x=3.248 y=0.939