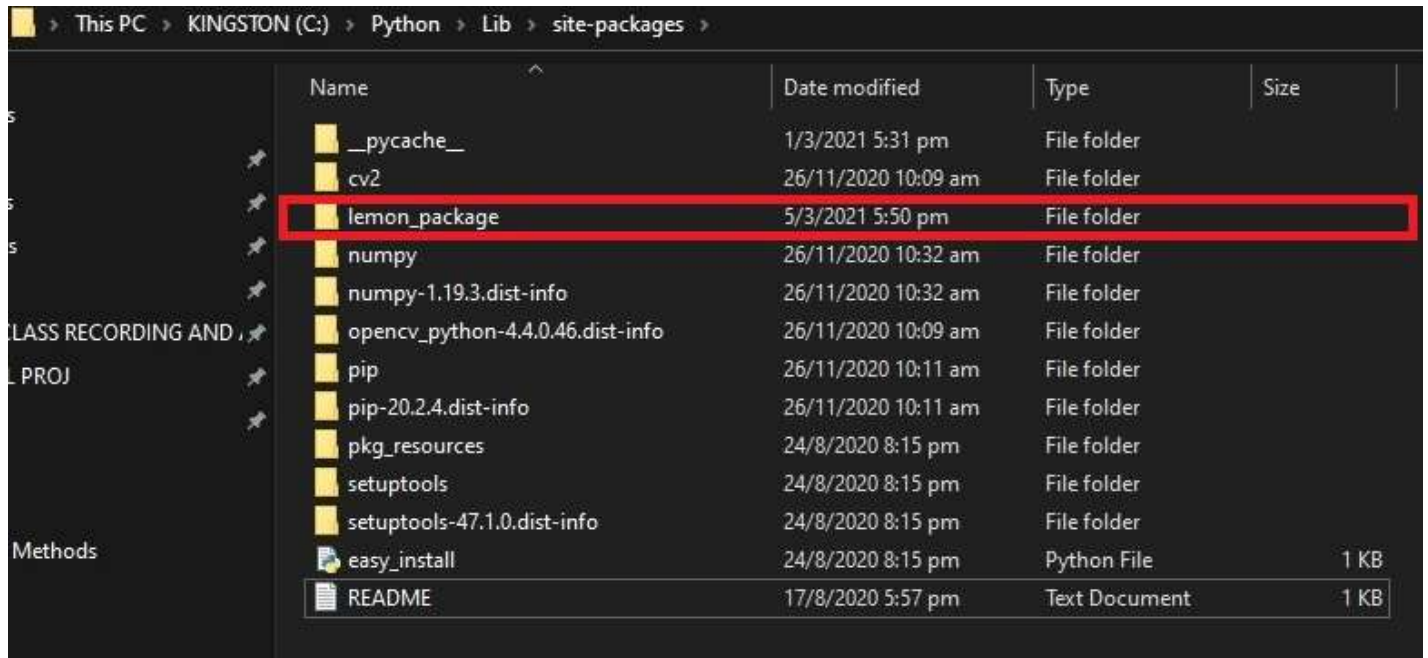
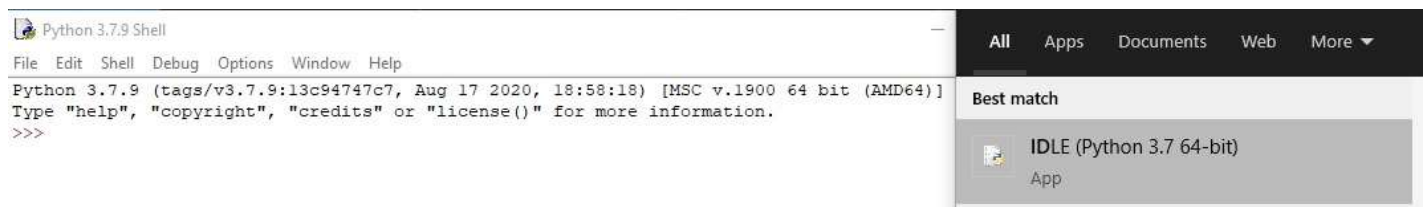


▼ Installing Lemon Package

- 1.) Download the lemon package in the GitHub Repository provided. [Lemon Package](#)
- 2.) Install the lemon package in the directory, Python\Lib\site-packages

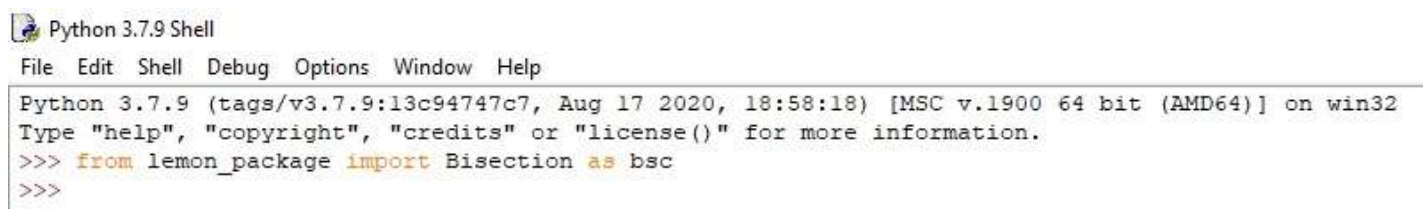


- 3.) Once the lemon package is installed, the lemon modules can now be used.
- 4.) To find the roots using the lemon modules, open the Python IDLE first.



▼ Using Bisection

- 1.) Import the necessary packages such as NumPy and Lemon Package.

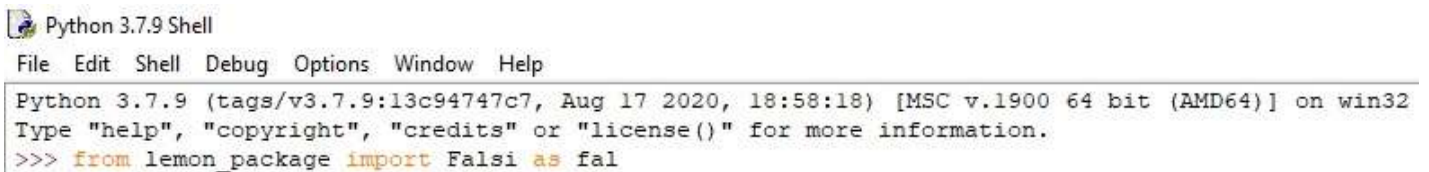


2.) Provide a function where the root will be found and provide the necessary parameters of the Bisection module. Try different intervals to find the most roots. Lastly, call the bisection method to output the roots of the function provided.

```
>>> func = lambda x: 2*x**4 + 3*x**3 - 11*x**2 - 9*x + 15
>>> root,end_bisect= bsc.bisection(func,i1 = -10,i2 = 1,n_roots = 5,epochs = 100,n_move = 50,tol = 1.0e-06)
>>> print("The roots are {}, at bisection: {}".format(root,end_bisect))
The roots are [-2.5 -1.732 1. 1.732], at bisection: 0
>>>
```

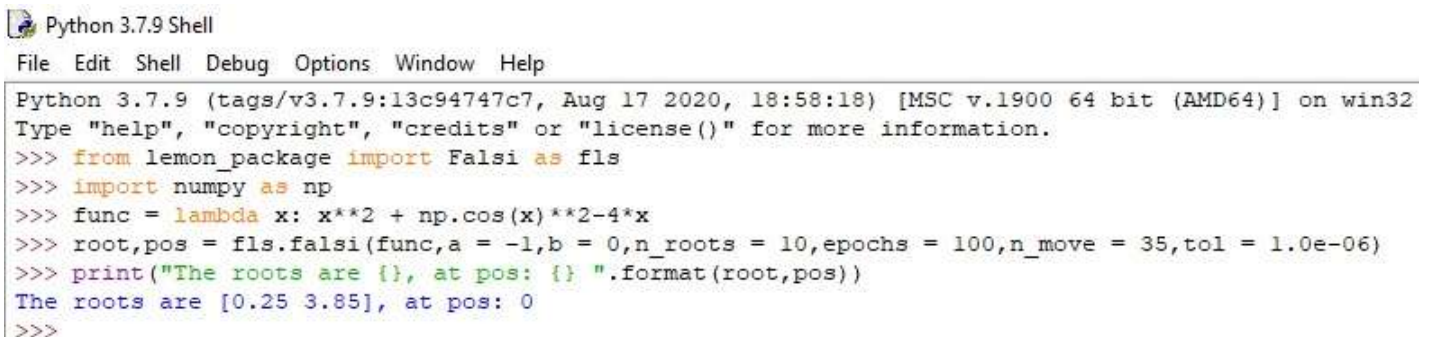
▼ Using Falsi

1.) Import the necessary packages such as NumPy and Lemon Package.



```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from lemon_package import Falsi as fal
```

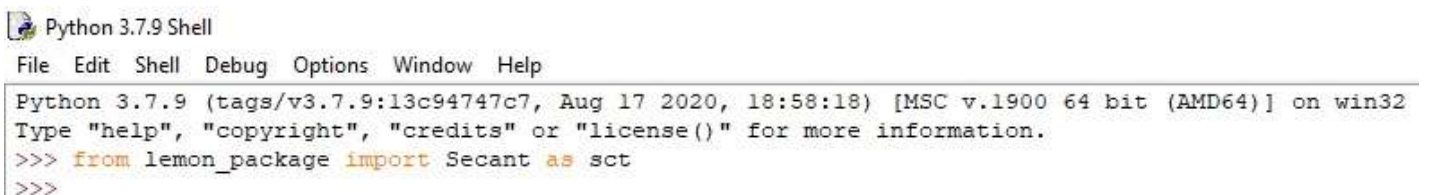
2.) Provide a function where the root will be found and provide the necessary parameters of the Falsi module. Try different intervals to find the most roots. Lastly, call the Falsi method to output the roots of the function provided.



```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from lemon_package import Falsi as fls
>>> import numpy as np
>>> func = lambda x: x**2 + np.cos(x)**2-4*x
>>> root,pos = fls.falsi(func,a = -1,b = 0,n_roots = 10,epochs = 100,n_move = 35,tol = 1.0e-06)
>>> print("The roots are {}, at pos: {}".format(root,pos))
The roots are [0.25 3.85], at pos: 0
>>>
```

▼ Using Secant

1.) Import the necessary packages such as NumPy and Lemon Package.



```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from lemon_package import Secant as sct
>>>
```

2.) Provide a function where the root will be found and provide the necessary parameters of the Secant module. Try different intervals to find the most roots. Lastly, call the Secant method to output the roots of the function provided.

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from lemon_package import Secant as sct
>>> func = lambda x: 2*x**4 + 3*x**3 - 11*x**2 - 9*x + 15
>>> root,end_epoch = sct.secant(func,a = -10,b = 1,n_roots = 4,epochs = 100,n_move = 40,tol = 1.0e-06)
>>> print("The roots are {}, at epoch: {}".format(root,end_epoch))
The roots are [-2.5 -1.732 1. 1.732], at epoch: 8
>>>
```

▼ Documentation of the Lemon Modules

To know more about the uses and syntax of the brute force and newton-rhapson module. Check its documentation in the link provided. [Python Package Manual](#)