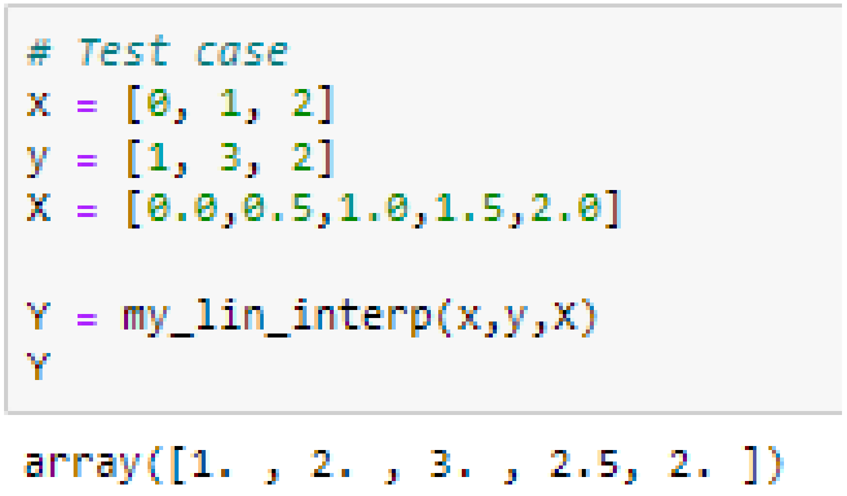
Practice – Week 07

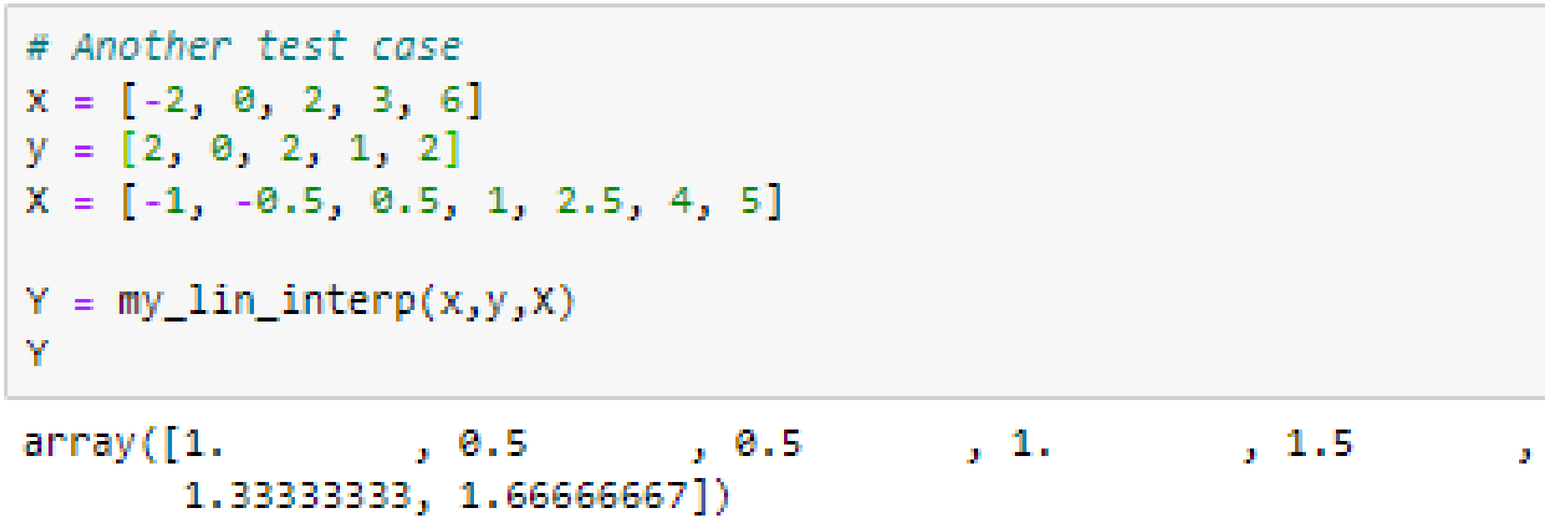
Practice – 01

1. Write a function **my_lin_interp(x, y, X)**, where **x** and **y** are arrays containing experimental data points, and **X** is an array. Assume that **x** and **X** are in ascending order and have unique elements. The output argument, **Y**, should be an array, the same size as **X**, where **Y[i]** is the **linear interpolation** of **X[i]**. You should **not** use **interp** from numpy or **interp1d** from scipy.



```
# Test case
x = [0, 1, 2]
y = [1, 3, 2]
X = [0.0,0.5,1.0,1.5,2.0]

Y = my_lin_interp(x,y,X)
Y
```

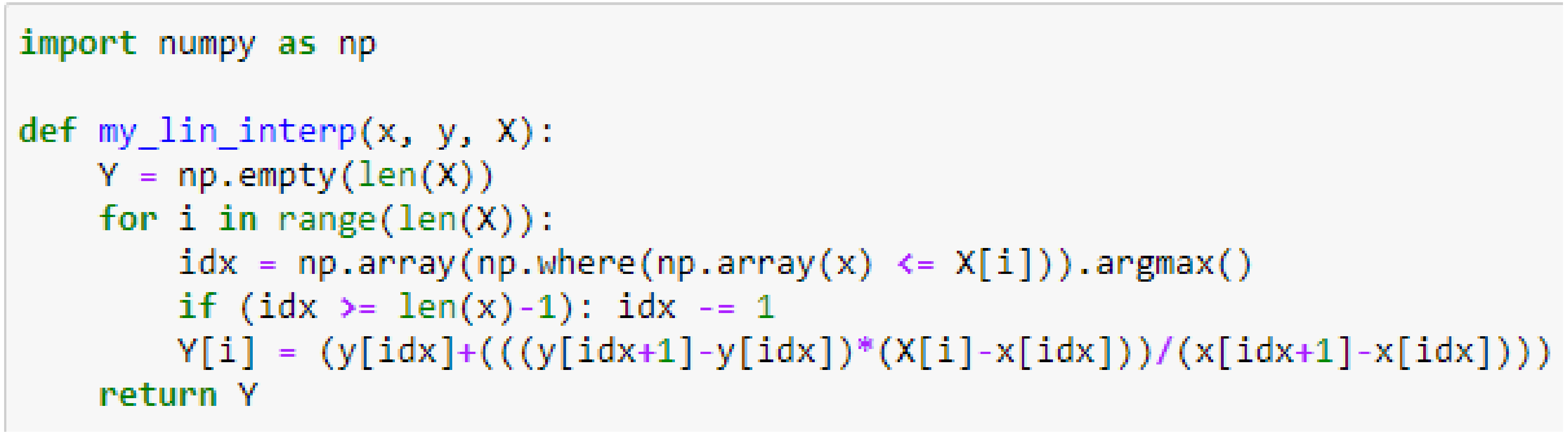
```
array([1. , 2. , 3. , 2.5, 2. ])
```



```
# Another test case
x = [-2, 0, 2, 3, 6]
y = [2, 0, 2, 1, 2]
X = [-1, -0.5, 0.5, 1, 2.5, 4, 5]

Y = my_lin_interp(x,y,X)
Y
```

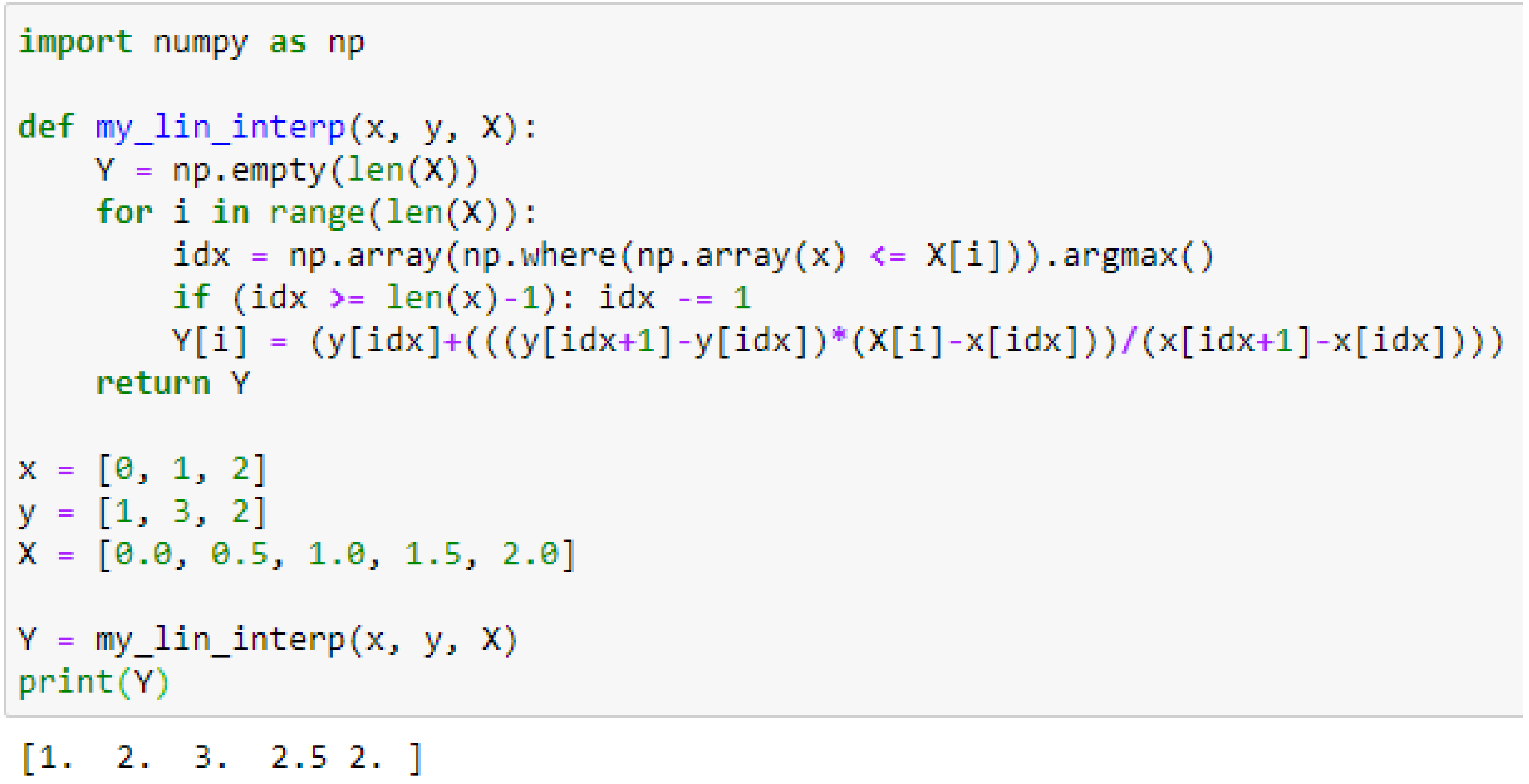
```
array([1.        , 0.5       , 0.5       , 1.        , 1.5       ,
       1.33333333, 1.66666667])
```

Practice – 01 – Answer



```python
import numpy as np

def my_lin_interp(x, y, X):
    Y = np.empty(len(X))
    for i in range(len(X)):
        idx = np.array(np.where(np.array(x) <= X[i])).argmax()
        if (idx >= len(x)-1): idx -= 1
        Y[i] = (y[idx]+(((y[idx+1]-y[idx])*(X[i]-x[idx]))/(x[idx+1]-x[idx])))
    return Y
```

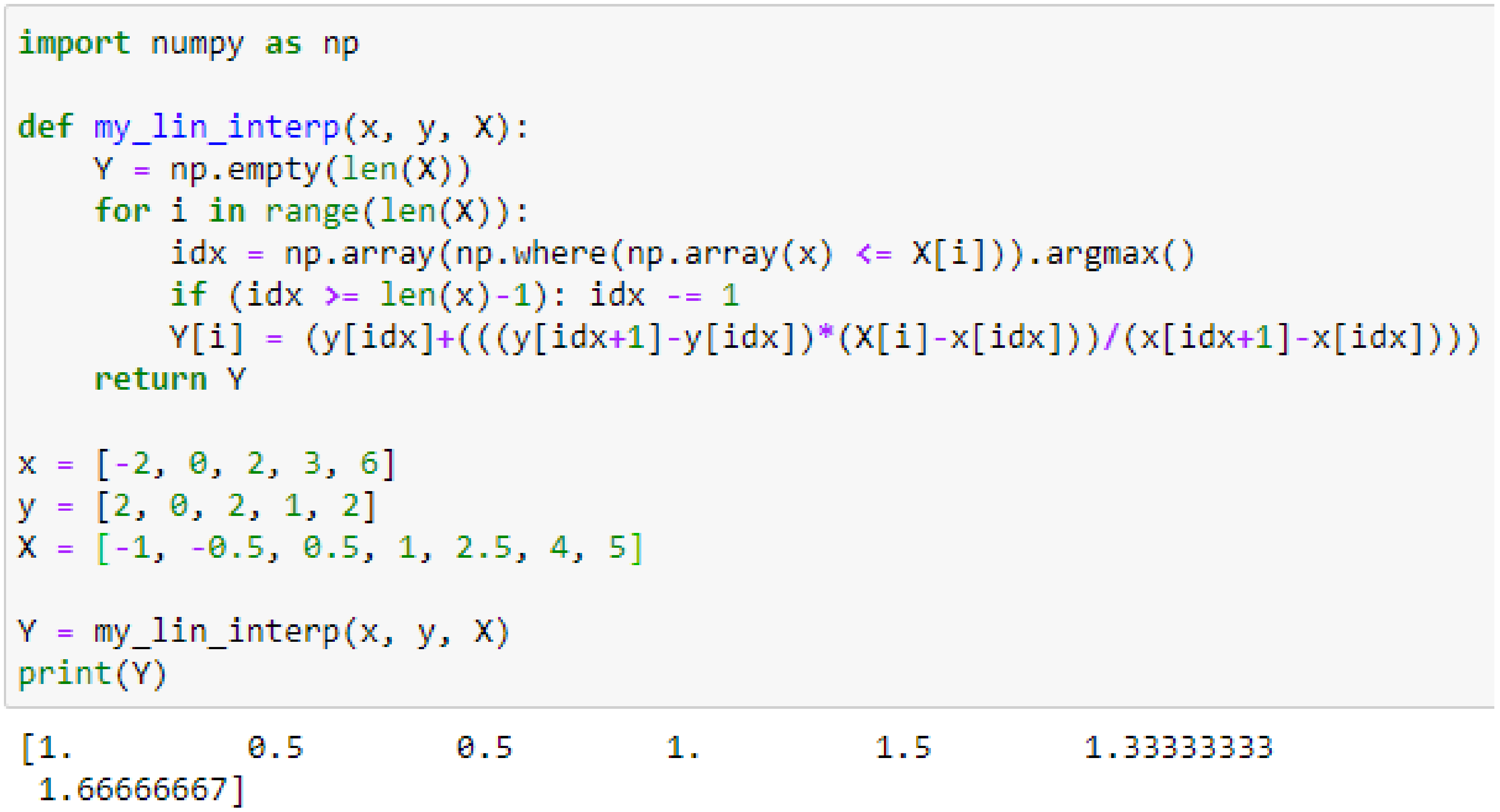
Practice – 01 – Answer



```python
import numpy as np

def my_lin_interp(x, y, X):
    Y = np.empty(len(X))
    for i in range(len(X)):
        idx = np.array(np.where(np.array(x) <= X[i])).argmax()
        if (idx >= len(x)-1): idx -= 1
        Y[i] = (y[idx]+(((y[idx+1]-y[idx])*(X[i]-x[idx]))/(x[idx+1]-x[idx])))
    return Y

x = [0, 1, 2]
y = [1, 3, 2]
X = [0.0, 0.5, 1.0, 1.5, 2.0]

Y = my_lin_interp(x, y, X)
print(Y)
```

```
[1.  2.  3.  2.5 2. ]
```

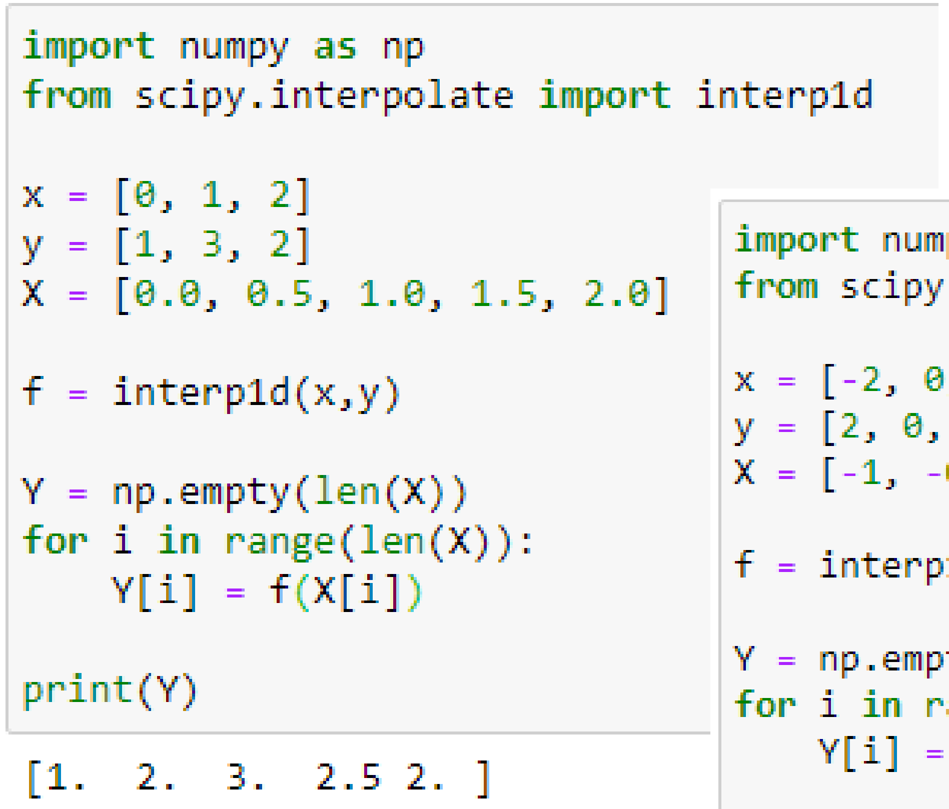
Practice – 01 – Answer



```python
import numpy as np

def my_lin_interp(x, y, X):
    Y = np.empty(len(X))
    for i in range(len(X)):
        idx = np.array(np.where(np.array(x) <= X[i])).argmax()
        if (idx >= len(x)-1): idx -= 1
        Y[i] = (y[idx]+(((y[idx+1]-y[idx])*(X[i]-x[idx]))/(x[idx+1]-x[idx])))
    return Y

x = [-2, 0, 2, 3, 6]
y = [2, 0, 2, 1, 2]
X = [-1, -0.5, 0.5, 1, 2.5, 4, 5]

Y = my_lin_interp(x, y, X)
print(Y)
```

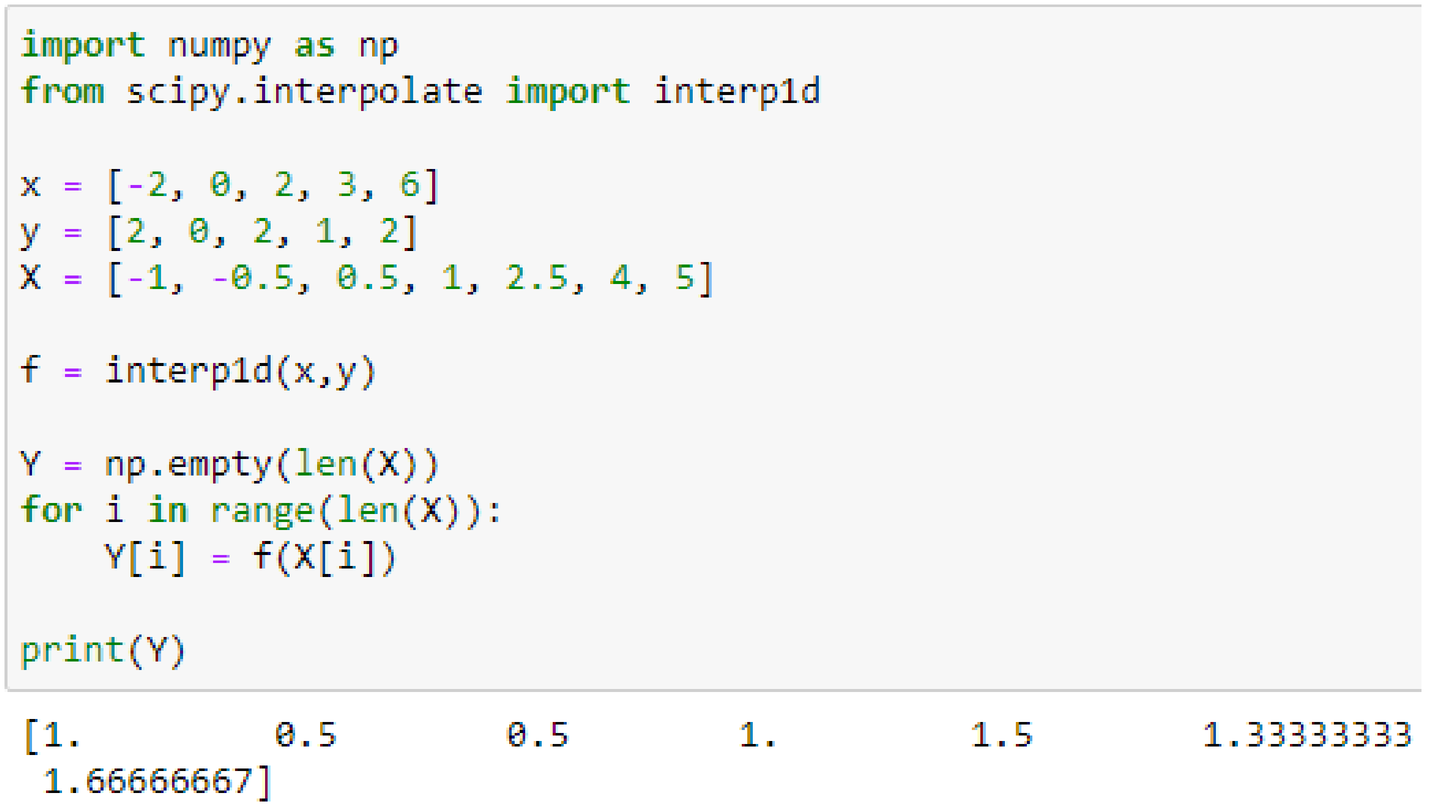
```
[1.         0.5        0.5        1.         1.5        1.33333333
 1.66666667]
```

Practice – 01 – Answer → Ref. interp1d



```
import numpy as np
from scipy.interpolate import interp1d

x = [0, 1, 2]
y = [1, 3, 2]
X = [0.0, 0.5, 1.0, 1.5, 2.0]

f = interp1d(x,y)

Y = np.empty(len(X))
for i in range(len(X)):
    Y[i] = f(X[i])

print(Y)
```

```
[1.  2.  3.  2.5 2. ]
```



```
import numpy as np
from scipy.interpolate import interp1d

x = [-2, 0, 2, 3, 6]
y = [2, 0, 2, 1, 2]
X = [-1, -0.5, 0.5, 1, 2.5, 4, 5]

f = interp1d(x,y)

Y = np.empty(len(X))
for i in range(len(X)):
    Y[i] = f(X[i])

print(Y)
```

```
[1.         0.5        0.5        1.         1.5        1.33333333
 1.66666667]
```