1. What is the result of the code, and explain?

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

* It will print ‘iNeuron’ as X is a global variable

2. What is the result of the code, and explain?

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

* It will print “iNeuron” as outer X is a global variable and inner X is local variable executing function will not change the value of outer X

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

* 'NI'
* 'iNeuron'
* Executing function will print the value of local variable and printing X will print value of global variable.

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

* “NI” will the output of this code as value of Global variable will be changed due to function.

5. What about this code—what’s the output, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

* It is hard tell without proper indentation it will either print iNeuron or keep on printing NI endlessly because if recursive function nested

6. How about this code: what is its output in Python 3, and explain?

>>> def func():

X = 'NI'

def nested():

nonlocal X

X = 'Spam'

nested()

print(X)

>>> func()

* Won’t print anything as nested function will be called recursively before reaching print function.