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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

Ans: iNeuron

In this code, a variable X is defined and assigned the string value 'iNeuron' outside of the func function. The function func does not have X defined locally, so it accesses the global variable X when it is called. When the func function is called, it prints the value of the global variable X, which is 'iNeuron'.

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

Ans: iNeuron

In this code, a global variable X is defined and assigned the string value 'iNeuron'. The function func attempts to assign the string value 'NI!' to a local variable X within the function's scope. When the function func is called, it sets a local variable X to 'NI!'. However, since this X is local to the function, it doesn't affect the value of the global variable X. After the function call, the print statement prints the value of the global variable X, which remains unchanged at 'iNeuron'.

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

Ans: NI

iNeuron

In this code, a global variable X is defined and assigned the string value 'iNeuron'. The function func attempts to assign the string value 'NI' to a local variable X within its scope. When the func function is called, it prints the value of the local variable X, which is 'NI'. However, since this X is local to the function, it doesn't affect the value of the global variable X. After the function call, the print statement prints the value of the global variable X, which remains unchanged at 'iNeuron'.

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

Ans: NI

In this code, a global variable X is defined and assigned the string value 'iNeuron'. The function func attempts to modify the global variable X using the global keyword, assigning the string value 'NI' to it. When the func function is called, it modifies the value of the global variable X to 'NI' using the global statement. After the function call, the print statement prints the modified value of the global variable X, which is now 'NI'.

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

Ans: NI

'iNeuron'

In this code, a global variable X is defined and assigned the string value 'iNeuron'. The func function attempts to assign the string value 'NI' to a local variable X within its scope. The nested function is defined within the func function and attempts to print the value of the local variable X defined in the func function. When the func function is called, it defines a local variable X with the value 'NI' but does not directly affect the global variable X. The nested function is then called within the func function, and it attempts to print the value of the local variable X within the func function's scope. However, since there is no local variable X in the nested function's scope, it looks for the variable in the enclosing scope, which is the global scope, and finds the global variable X with the value 'iNeuron'. After the function calls, the value of the global variable X is returned as the final line of the code.

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()

Ans: Spam

In this code, the func function attempts to assign the string value 'NI' to a local variable X within its scope. The nested function is defined within the func function and attempts to modify the variable X declared in the enclosing scope of the func function using the nonlocal keyword. When the func function is called, it defines a local variable X with the value 'NI'. The nested function is then called within the func function, and it modifies the variable X using the nonlocal keyword to change its value to 'Spam'. After the nested function call, the value of the modified local variable X is printed within the func function's scope.