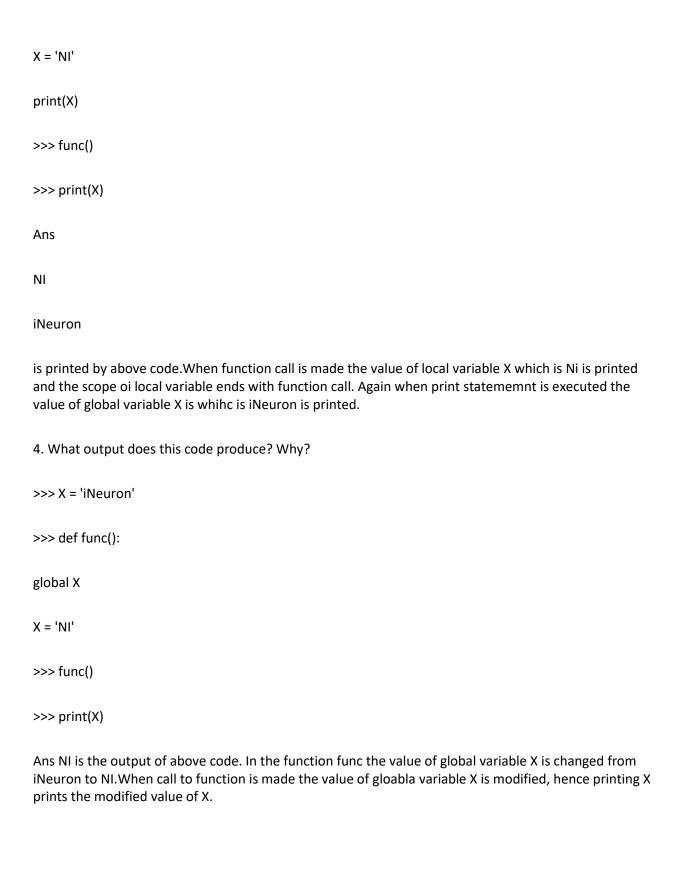
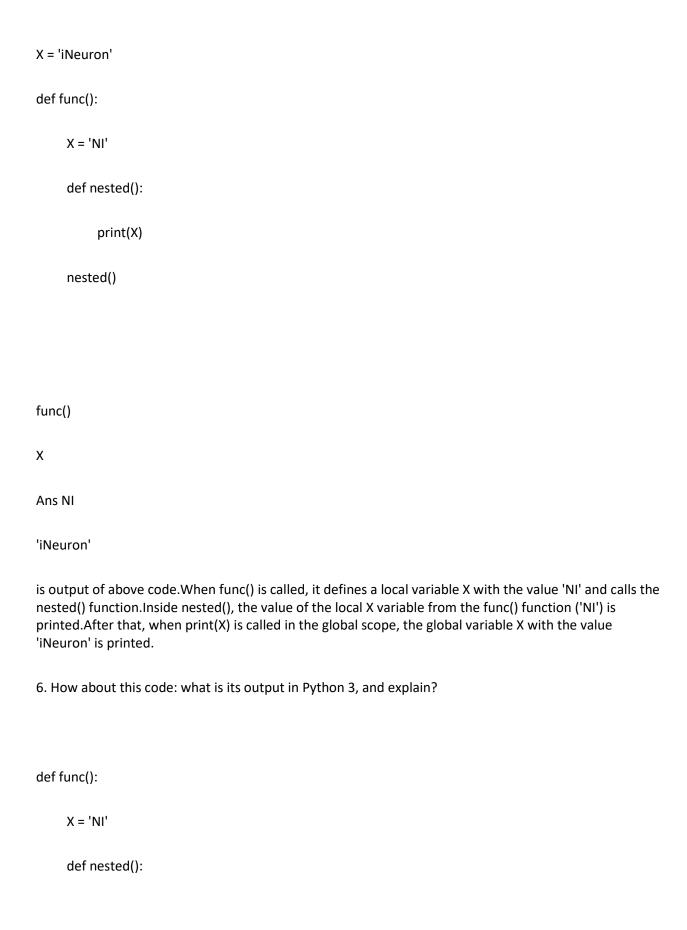
1. What is the result of the code, and explain?
>>> X = 'iNeuron'
>>> def func():
print(X)
>>> func()
Ans iNeuron is the output of above code because X is global variable and global variables can be accessed within the function
2. What is the result of the code, and explain?
>>> X = 'iNeuron'
>>> def func():
X = 'NI!'
>>> func()
>>> print(X)
Ans iNeuron is the output of above code . In this case 2 variables with the name X are defined. The variable X = 'iNeuron' is global variable and X = 'NI!' is local variable. The scope of local variable ends when the function call ends. Hence printing X prints the value of global variable X having value iNeuron.
3. What does this code print, and why?
>>> X = 'iNeuron'
>>> def func():



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



```
nonlocal X

X = 'Spam'

nested()

print(X)
```

Ans Spam is output of above code. The nonlocal keyword allows the nested() function to modify the variable X in the nearest enclosing scope, which is the func() function's scope.