1. The output of this code will be "iNeuron". This is because the variable X is defined outside of the function func(), and therefore is accessible inside the function.

2. The output of this code will be "iNeuron". This is because the variable X is defined outside of the function func(), and the line "X = 'NI!'" inside the function creates a new local variable X that shadows the global variable X.

3. The output of this code will be "NI". This is because the line "X = 'NI'" inside the function creates a new local variable X that shadows the global variable X, and the print statement inside the function prints the value of the local variable X.

4. The output of this code will be "NI". This is because the "global X" statement inside the function func() tells Python to use the global variable X instead of creating a new local variable with the same name, and the line "X = 'NI'" changes the value of the global variable X.

5. The output of this code will be "NI". This is because the line "nested()" calls the nested function, which prints the value of the variable X, and the nested function has access to the variable X defined in the enclosing scope of the function func().

6. The output of this code will be "Spam". This is because the "nonlocal X" statement inside the nested function tells Python to use the variable X defined in the enclosing scope of the function func(), and the line "X = 'Spam'" inside the nested function changes the value of that variable. When the function func() is called, it calls the nested function, which changes the value of X to "Spam", and then prints the new value of X.