

Question 1

Running styleLint.py ended with an error.

```
Last login: Tue Apr 12 18:59:32 2022 from 192.168.11.51
codio@rivergalileo-japandallas:~/workspace$ python3 styleLint.py
File "styleLint.py", line 5
    """ Return factorial of n """
        ^
IndentationError: expected an indented block
```

To fix this, the code needed to have tabbed indents in the function and if blocks.

```
1
2 # CODE SOURCE: SOFTWARE ARCHITECTURE WITH PYTHON
3
4 def factorial(n):
5     """ Return factorial of n """
6     if n == 0:
7         return 1
8     else:
9         return n*factorial(n-1)
```

Question 2

Running pylint on pylintTest.py results in the following feedback:

```
codio@rivergalileo-japandallas:~/workspace$ pylint pylintTest.py
***** Module pylintTest
pylintTest.py:5:0: C0303: Trailing whitespace (trailing-whitespace)
pylintTest.py:12:0: W0311: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
pylintTest.py:13:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
pylintTest.py:14:0: W0311: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
pylintTest.py:15:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
pylintTest.py:16:0: W0311: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
pylintTest.py:17:0: W0311: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
pylintTest.py:18:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
pylintTest.py:19:0: W0311: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
pylintTest.py:20:0: W0311: Bad indentation. Found 8 spaces, expected 16 (bad-indentation)
pylintTest.py:21:0: W0311: Bad indentation. Found 12 spaces, expected 20 (bad-indentation)
pylintTest.py:22:0: W0311: Bad indentation. Found 8 spaces, expected 16 (bad-indentation)
pylintTest.py:23:0: W0311: Bad indentation. Found 10 spaces, expected 20 (bad-indentation)
pylintTest.py:24:0: W0311: Bad indentation. Found 10 spaces, expected 20 (bad-indentation)
pylintTest.py:26:0: C0304: Final newline missing (missing-final-newline)
pylintTest.py:1:0: C0103: Module name "pylintTest" doesn't conform to snake_case naming style (invalid-name)
pylintTest.py:1:0: C0114: Missing module docstring (missing-module-docstring)
pylintTest.py:6:0: C0103: Constant name "shift" doesn't conform to UPPER_CASE naming style (invalid-name)
pylintTest.py:7:9: E0602: Undefined variable 'raw_input' (undefined-variable)
pylintTest.py:8:8: E0602: Undefined variable 'raw_input' (undefined-variable)
pylintTest.py:9:0: C0103: Constant name "letters" doesn't conform to UPPER_CASE naming style (invalid-name)
pylintTest.py:10:0: C0103: Constant name "encoded" doesn't conform to UPPER_CASE naming style (invalid-name)
pylintTest.py:14:6: C0103: Constant name "encoded" doesn't conform to UPPER_CASE naming style (invalid-name)
pylintTest.py:16:6: C0103: Constant name "x" doesn't conform to UPPER_CASE naming style (invalid-name)
pylintTest.py:17:6: C0103: Constant name "encoded" doesn't conform to UPPER_CASE naming style (invalid-name)
pylintTest.py:19:6: W0621: Redefining name 'letter' from outer scope (line 12) (redefined-outer-name)
pylintTest.py:21:12: C0103: Constant name "encoded" doesn't conform to UPPER_CASE naming style (invalid-name)
pylintTest.py:23:10: C0103: Constant name "x" doesn't conform to UPPER_CASE naming style (invalid-name)
pylintTest.py:24:10: C0103: Constant name "encoded" doesn't conform to UPPER_CASE naming style (invalid-name)

-----
Your code has been rated at 0.00/10
```

After addressing all of the issues

```
codio@rivergalileo-japandallas:~/workspace$ pylint pylint_test.py
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
codio@rivergalileo-japandallas:~/workspace$
```

Question 3

Running flake8 on pylintTest, the feedback is much less detailed.

```
codio@rivergalileo-japandallas:~/workspace$ flake8 pylintTest_original.py
pylintTest_original.py:25:13: E999 SyntaxError: invalid syntax
codio@rivergalileo-japandallas:~/workspace$
```

After adding parentheses around encoded variable in the print command, the feedback was more detailed, but still less detailed than pylint.

```
codio@rivergalileo-japandallas:~/workspace$ flake8 pylintTest_original.py
pylintTest_original.py:25:13: E999 SyntaxError: invalid syntax
codio@rivergalileo-japandallas:~/workspace$ flake8 pylintTest_original.py
pylintTest_original.py:4:1: W293 blank line contains whitespace
pylintTest_original.py:6:10: F821 undefined name 'raw_input'
pylintTest_original.py:7:9: F821 undefined name 'raw_input'
pylintTest_original.py:11:3: E111 indentation is not a multiple of 4
pylintTest_original.py:13:7: E111 indentation is not a multiple of 4
pylintTest_original.py:15:7: E111 indentation is not a multiple of 4
pylintTest_original.py:15:35: F821 undefined name 'shift'
pylintTest_original.py:16:7: E111 indentation is not a multiple of 4
pylintTest_original.py:16:14: E225 missing whitespace around operator
pylintTest_original.py:18:7: E111 indentation is not a multiple of 4
pylintTest_original.py:22:11: E111 indentation is not a multiple of 4
pylintTest_original.py:22:39: F821 undefined name 'shift'
pylintTest_original.py:23:11: E111 indentation is not a multiple of 4
pylintTest_original.py:25:15: W292 no newline at end of file
codio@rivergalileo-japandallas:~/workspace$
```

Question 4

Running mccabe on sums2.py resulted in the following complexity scores:

```
python -m mccabe mccabe.py
```

```
codio@rivergalileo-japandallas:~/workspace$ python3 -m mccabe sums2.py
4:0: 'test_sum' 1
7:0: 'test_sum_tuple' 1
If 10 2
codio@rivergalileo-japandallas:~/workspace$
```

The mccabe module counts the possible paths through the code from decisions.