420-PRO-LCU Programming in Python - Lab 13 Final Review

May 5, 2022

1. True or False

   (a) Given the function header: def g(x = 0, y = 2):, the function call g() is equivalent to the function call g(0, 2)

   (b) The following expression would yield the value True:
   list(zip([1, 9], [4, 10])) == [[1, 4], [9, 10]]

2. What would each of the following 3 Python code fragments print?

   (a)
```
x, y, z = 3, 1, 2
try:
    z = z + 1
    y = z / x
    x = z - 1
except ZeroDivisionError:
    y = 9
except Exception:
    y = 10
else:
    y = 20
print(x, y, z)
```

   (b)
```
x, y, z = 0, 1, 2
try:
    z = z + 1
    y = z / x
    x = z - 1
except Exception:
    y = 10
except ZeroDivisionError:
    y = 9
else:
    y = 20
print(x, y, z)
```

   (c)
```
x, y, z = 0, 1, 2
try:
    z = z + 1
    y = z / x
    x = z - 1
except ZeroDivisionError:
    y = 9
except Exception:
    y = 10
else:
    y = 20
print(x, y, z)
```

3. What would each of the following Python code fragments print?

   (a)
```
result = ''
for ch in "abcdef"[1:]:
    result = ch + result
result = '/' + result + '/'
print(result)
```

   _____

   (b)
```
x, y, z = False, False, True
print(x and y or z, z and not y or x)
```

   _____

   (c)
```
a, b, c = 'False', '', 0
print(not a or not b or not c)
```

   _____

   (d)
```
x, y, z = [], [0], True
print(x and y[0] or z, z and not y or bool(x))
```

   _____

   (e)
```
text = 'To be or not to be'
print(text.split()[::-2])
```

(f)
```
array = [5,4,3,2,1]
s = 0
for n in array[::2]:
  s += n
print(s)
```

(g)
```
def f(x):
   return x // 2

def g(x, y):
   return x ** y

print(f(g(3, 3)))
```

(h)
```
y = []
for v in range(0, 30, 5):
   y.append(v % 2 != 0)
print(all(y))
```

4. Write a function `mean_dict` that takes two dictionaries as arguments and uses them to construct a third dictionary. The output dictionary should contain all of the elements of both dictionaries, but if the same key is present in both arguments, the output dictionary should contain the average of the two input values. Exclude any item with a zero value, and *do not modify* the input arguments! For example, with the inputs:

```
x = {'a': 2, 'b': 1, 'c': 5}
y = {'a': 3, 'b': -1, 'c': 3, 'd': 3 }
the returned dictionary should be: {'a': 2.5, 'c': 4.0, 'd': 3}
```

5. Using recursion, write a function `recursive_power(n,x)` that returns $n**x$.

```
print(recursive_power(3,4)) #prints 81
```

6. Write a function `get_diagonal(m)` that returns the diagonal elements of a square matrix represented as a list. For example, for the matrices:

```
mat1 = [[1, 2, 3], [5, 4, 6], [9, 7, 8]]
mat2 = [[1, 2, 3], [5, 4, 6], [9, 7, 8], [10, 12, 11]]
```

the call `get_diagonal(mat1)` would return the list: [1, 4, 8]
and the call `get_diagonal(mat2)` would display the message "Not a square matrix "