

CS351-Cloud Computing

Assignment 1

Python Warmup Assignment

Due on 10 August 2022 (02.59PM)

Solve the following problems in Python 3.6+ .

In addition to making sure that your code gives the correct solution, each solution must meet the following conditions.

1. Your function parameters must be [type-hinted] (<https://docs.python.org/3.6/library/typing.html>). These need not be perfect, but should be generally correctly describe your input and output.
2. Your functions must have a doc-string describing what the function does. Keep is succinct and concise.
3. Use a auto-formatter such as [black] (<https://pypi.org/project/black/>) or [autopep8] (<https://pypi.org/project/autopep8/>).
4. Write atleast 2 simple [doctests] (<https://docs.python.org/3.6/library/doctest.html>) per solution to verify that your code is working correctly.
5. Use descriptive variable names.

Here is an example of what your code should look like.

For example

```
```python
from typing import List
```

```
def average(num_list: List[int]) -> float:
 """Finds the average of a list of numbers.
```

```
 doctests:
 >>> average([1,2,3,4,5])
 3.0
 >>> average([0])
 0.0
 """
 list_sum = sum(num_list)
 list_size = len(num_list)
 avg = list_sum/list_size
 return avg
```

```
if __name__ == "__main__":
 import doctest
 doctest.testmod()
...

```

## Problem 1

Write a function to break down a string into a list of characters.

```

'''
Input: "abc"
Output: ['a','b','c']
'''

Problem 2
Write a function to reverse output of the problem 1 back into a string
'''
Input: ['a','b','c']
Output: "abc"
'''

Problem 3
Write a function generate a list of n random numbers.
Use the inbuilt `random` module.
'''
Input: 5
Output: [5,2,3,1,5]
'''

Problem 4
Write a function a sort a given list of numbers in descending order.
'''
Input: [1,2,3,4,5]
Output: [5,4,3,2,1]
'''

Problem 5
Write a function to get frequency of each numbers in a list of numbers.
Use a python `dict` to solve this.
'''
Input: [1,1,3,2,3,2,3,2,2]
Output: {1: 2, 3: 3, 2: 4}
'''

Problem 6
Write a function to get all the unique elements from given list.
Your solution must use `set` to solve this.
'''
Input: [1,1,3,2,3,2,3,2,2]
Output: {1,2,3}
'''

Problem 7
Write a function to get the first repeating element from list. Your
solution must use `set` to solve this.
'''
Input: [1,2,3,4,5,1,2]
Output: 1
'''

Problem 8

```

Write a function that takes an integer n and output a `dict` containing keys from 0,2 ... to n and each key is mapped to a list containing the square and cube of the number.

```
...
Input: 3
Output:
{
 0:[0,0],
 1:[1,1],
 2:[4,8],
 3:[9,27]
}
...
```

#### ## Problem 9

Given two lists of equal size, write a function to create tuples of each consecutive element having same index. Use `zip` in some capacity to solve this.

```
...
Input: [1,2,3,4], ['a','b','c','d']
Output: [(1,'a'), (2,'b'), (3,'c'), (4,'d')]
...
```

#### ## Problem 10

Write a function that uses list comprehension to generate the squares of 0 to n.

```
...
Input : 5
Output : [0, 1, 4, 9, 16, 25]
...
```

#### ## Problem 11

Write a function that uses dictionary comprehension to generate a mapping from (0 to n) to their squares.

```
...
Input : 5
Output : {0:0, 1:1, 2:4, 3:9, 4:16, 5:25}
...
```

#### ## Problem 12.

Write a `class` such that :

1. The initializer takes an arbitrary list of atomic values as input and saves it in a instance variable.
2. Has a method called `apply` which has the following functionality:
  1. Accepts a function as a parameter. You can use a lambda function.
  2. Applies the function to saved list and return the output. The instance variable must not be modified.
  3. If it fails `raise` an `Exception` with a custom error message. You can use `try` and `except` here.

```
```python
def sq(x):
    return x**2
```

```
c1 = MyClass([1,2,3,4])

print(c1.apply(lambda x:x**2))
[1,4,9,16]
```

```
c2 = MyClass(['a','b','c'])
c2.apply(sq)
```

```
-----
-----
TypeError                                Traceback (most recent call
last)
```

```
....
....
Exception: Custom Error
\`\`\`
```

```
## Problem 13
```

Write a function takes as input a list of words and upper-cases each word. Use ``functools.map`` in some capacity to solve this.

```
```
Input : ['aa','bb','cd','e']
Output : ['AA', 'BB', 'CD', 'E']
\`\`\`
```

```
Problem 14:
```

Write a function to find the product of all the numbers in a list using ``functools.reduce`` in some capacity.

```
```
Input : [1,2,3,4,5]
Output : 120
\`\`\`
```