Programming I (Python)
Assignment 3

Instructions

- Similar to Assignment 2.
- Some questions may have multiple input/output variable. Please watch.

Questions

1. You have been given a list that has elements which occur more than once in that list. Iterate over the elements of the list and make a dictionary which contains the keys as the elements and their count as the value of the key.

   For example, for `inp = [1, 1, 3, 2, 1, 2, 3, 2, 2, 2]`, the answer should be `output = {1: 3, 2: 5, 3: 2}`.

2. (a) You have 2 dictionaries, which may or may not have overlapping keys. Create a new dictionary. If any key is common to the dictionaries, the value in the new dictionary will be the sum of the values of the individual dictionaries, otherwise just add the key-value pair in the new dictionary. For example:

   ```
   Dic1 = {'some': 5, 'fuzzy':25, 'data':15, 'logic':0}
   Dic2 = {'some': 15, 'data':2, 'marks':100}
   ```

   Then, the answer should be:

   ```
   output = {'some':20, 'fuzzy':25, 'data':17, 'logic':0. 'marks':100}
   ```

   (b) (*challenge; not to be submitted*) The same problem as above but instead of just 2 dictionaries, the number of dictionaries is arbitrary.

3. Given a string, write a program to find whether the given string is a heterogram or not.

   Reference: https://en.wikipedia.org/wiki/Heterogram_(literature)

   (Think of a data structure in Python which can do this for you)

4. Write a program to push all the zeroes in a list to the end of the list. Use list comprehension for this. The order of occurrence of all the other characters should be preserved. For example: for `inp = [1,0,2,5,0,7]` the answer should be `output = [1,2,5,7,0,0]`.

5. Given two sets of lengths m and n respectively, find how many complete strings are there if any two elements of these sets are concatenated. (Complete strings are the strings which contain all the 26 characters of the English alphabet). For example: `set1 = {'abcdefgh', 'abcd', 'wxyz'}` `set2 = {'ijklmnopqrstuvwxyz', 'abefgh'}`, there are 6 possible pairs:

    1. `abcdefghijklmnopqrstuvwxyz`
    2. `abcdefghabefgh`
    3. `abcdijklmnopqrstuvwxyz`
    4. `abcdabefgh`
    5. `wxyzijklmnopqrstuvwxyz`
    6. `wxyzabefgh`

    Out of these 6, only first one is the complete string. So, your program should return answer as 1. Therefore, for the input:

```
set1 = {'abcdefgh', 'abcd', 'wxyz'}
set2 = {'ijklmnopqrstuvwxyz', 'abefgh'}
```

    Output should be (note that there are two output variables: `sets` and `output`.):

```
sets = [
  "abcdefghijklmnopqrstuvwxyz",
  "abcdefghabefgh",
  "abcdijklmnopqrstuvwxyz",
  "abcdabefgh",
  "wxyzijklmnopqrstuvwxyz",
  "wxyzabefgh"
]

output = 1
```

6. Given a string and an integer N, find the Nth most frequently occurring character in the string.

    1. If two or more elements have the same count, then return the character with the lowest ASCII value.
    2. There could be spaces in the string as well. You need to take care of that as well.

    For example: if string is `inp = 'pythonissuchaeasylanguage'` and `N = 3`, your answer should be `output = 'e'`.