

# Instructions

---

- There is no inherent order in these exercises , you can pick them in any order and work on them
  - These are not a test of what you have learnt but a tool to do couple things
    - To make you explore few ideas which were possibly not discussed in the class in too many details
    - To make you develop capability of connecting isolated ideas to solve a programming problem
    - To make you identify gaps in your understanding
  - Its perfectly fine if you are not able solve all or any of these problems , more important is that you tried and now you know what you need to work on to get better
  - Feel free to discuss these in the QA forum
  - It will be good for you , if you discuss the problems before looking at the uploaded solutions for the same
  - Uploaded solutions are one way of solving the problems, they are definitely not the only way [or the only right way]. You used some other method to solve the problem; thats ok too; but do discuss you solutions to see if there are some gaps in them.
  - Uploaded solutions might look neater and tidier than your solutions, which itself is a result of many iterations. What you finally get to see there is after a lot of cleanups and rewrites. Dont be too hard on yourself, what matters is that you tried solving or solved a problem .
- 

1. Generate a list containing 50 random integers between 10 to 100. Write a program which removes all the values which are divisible by 3 from the list . Then it subtracts 1 from all the remaining values and then removes all the values divisible by 5 from the list .

*Additional Info :*

*Example with a smaller list of random integers :*

```
starting list with random integers = [23,34,45,56,17,18,23,10]
after removing all the values divisible by 3 = [23,34,56,17,23,10]
subtracting 1 from each of the value = [22,33,55,16,22,9]
final output after removing values divisible by 5 = [22,33,16,22,9]
```

Way to create a list with random integers

```
import numpy as np
x=np.random.randint(10,101,50)
```

since these values are being generated randomly , your solution will not match exactly with the solution that anyone else comes up with [in terms of values that you see as outcome]

---

2. Consider the following dictionary

```
{'a':'e','b':'$', 'c':'q', 'd':'p', 'j':'n'}
```

Write a program which takes input any string and then replaces all the letters which are keys in the above dictionary with their corresponding values in the dictionary.

```
example input = 'abjected'
output = 'e$neqtep'
```

---

3. Write a program which takes input a list of strings and removes strings until all the remaining strings are of equal lengths . Assume that in the input list , there will be at-least two strings of the same length . For simplicity , assume that list will **not** contain equal length strings for multiple lengths

```
example input = ['abc', 'abcd', 'acbde', 'acbgeh', 'qwert']
output = ['abcde', 'qwert']
```

*\*Additional Info :\**

*\*=>you can identify for what length you have multiple strings by first creating a list containing lengths of the string and then look at count of the lengths in that list . any value which has count more than 1 , thats your target length. From the input list you can remove all other strings and keep only that string which correspond to that length .\**

*\*=> you can get unique values in a list x by using set(x)\**

*\*=> you can get count of a value , say `y` in list `x` using `x.count(y)` \**

---

4. Write a function which tells you whether you can take jugs of capacity 3 and 5 litres and fill another jug of capacity  completely with water using just these two . For simplicity assume that you can only pour into the jug of capacity  . You can not take any water out. And you start with jug of capacity  being empty. It is acceptable if you dont use any one of the jugs [of capacity 3 and 5 litres]

*Additional Details :*

=> function will take argument `x` , the capacity of the jug to be filled

=> The problem is same as finding if there exist positive integer solutions for  $a$  and  $b$  in the equation  $3a + 5b = x$  where  $x$  is a known integer

=>one of the strategies to find this out can be as per following example [dont be intimidated by the example, i have given multiple elaborate examples to explain possible scenarios to you ]

```
you start with values of a in the range [0 to int(x/3)] . for each of these
values of a that you are trying find the corresponding value of b = (x-3a)/5 .
if you get a value of b which is +ve integer [or 0], which means there exist a
+ve integer solution for the pair (a,b).
```

```
for example if x is 16 . we'll try values of a from 0 to [16/3]=5, we'll stop
early if we find a +ve integer solution for b. if we exhaust all the values of
a to be tried and dont find a solution, we'll output 'NO'
```

```
for a=0 , b = 16/5 => not an integer => rejected
a=1 , b =13/5 => not an integer => rejected
a=2 , b= 10/5=2 => a positive integer => accepted => output 'yes'
```

```
another example : if x is 17 [values of a to be tried 0,1,2,3,4,5]
```

```
for a=0 , b = 17/5 => not an integer => rejected
a=1 , b=14/5 => not an integer => rejected
a=2 , b=11/5 => not an integer => rejected
a=3 , b=8/5 => not an integer => rejected
a=4 , b=5/5=1 => a positive integer => accepted => output 'yes'
```

```
another example : if x is 7 [values of a to be tried 0,1,2]
```

```
for a=0 , b = 7/5 => not an integer => rejected
a=1 , b=4/5 => not an integer => rejected
a=2 , b=1/5 => not an integer => rejected
```

```
we ran out of values of a to be tried => outcome 'No'
```

=> this does not imply that this is the only possible way to solve this , feel free to use your own ideas if you have one

=> you can check if a number `x` is integer or not using `x==int(x)`

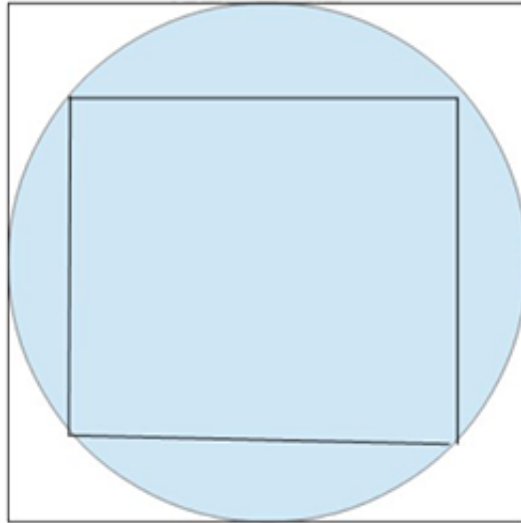
- 
5. Use dictionary comprehension to create a dictionary with keys : `t1,t2,t3.....t100` which takes random integers from the range `[1,100]` as their corresponding values. Write a function which takes this dictionary as input and returns number of `unique values` from all `key:value` pairs in that dictionary .

*Additional info:*

```
example input with a smaller dictionary : {'t1':2, 't2':45, 't3':34, 't4':2}
output = 3
```

---

6. Imagine a circle and two squares: a smaller and a bigger one. For the smaller one, the circle is a circumcircle and for the bigger one, an incircle.



Create a function, that takes an integer (radius of the circle) and returns the difference of the areas of the two squares.

---

7. A number is said to be **Harshad** if it's *exactly divisible* by the **sum** of its digits. Create a function that determines whether a number is a Harshad or not. Input to the function will be an integer.
- 

8. Write a function which takes an integer `n` as input and returns sum of all even numbers in the first `n` terms of a fibonacci series .

```
example input = 7
fibonacci series for the same will be : 1,1,2,3,5,8,13
output=2+8=10
```

---

*Additional info :* Each term in the fibonacci series is sum of previous two. First two terms of the series are 1 . For simplicity , you can assume that input `n` will always be greater than 4.

---

9. Write a function which takes input an english sentence of any length , and a set containing some stop words. Function then removes all stop words from the sentence and returns a dictionary where keys are all unique words in the sentence and values are their count in the sentence .

```
example input sentence = 'we must indeed all hang together or most assuredly we  
shall all hang separately'  
example input for stop word set = {'we','or','all','shall'}  
output =  
{ 'must':1, 'indeed':1, 'hang':2, 'together':1, 'most':1, 'assuredly':1, 'separately':  
1 }
```

*Additional info :* For simplicity assume that there will be no punctuations in the input sentence, simply words separated by single white spaces. Assume that all string inputs will be lowercase.

---

10. consider this list of addresses

['H-73, MDT, Powai , Mumbai' , '1604, SS, Hyderabad' , 'B block 73, Adyar, Chennai']

Extract cities from each address, use list comprehension .

Hints :

- Write list comprehension to convert this to list of lists [ use split function on strings]
  - Extract last element of each of these individual list using index -1, using list comprehension
- 

11. given a list , create a dictionary with counts of unique elements that it contains input list =  
['a','b','a','a','b','c','c','a','b'] output dictionary = {'a':4,'b':3,'c':2} Hints :

- iterate over the list using a for loop
  - for creating an empty dictionary : dict\_name={}
  - for adding a new element to dictionary: dict\_name['element']=value
  - for getting keys of dictionary : dict\_name.keys()
  - for checking if an element is in the dictionary : element in dict\_name.keys()
  - for incrementing the value assigned to an element in a dictionary : dict\_name['element']+=1
-