

# Open Optical Networks - Python Exercises

These exercises sets covers some aspects you will find useful for the final exam software development. These exercises are NOT part of the final exam and their goal is to make you get used to Python programming. All sets from 1 to 6 are provided with solutions. However, you are strongly encouraged to find yourself a solution to the presented problems and take the solution just as a guide. A final exercise is provided without solution in order to test what you have learned.

## Exercise Set 1

Python basics

Refer to <https://docs.python.org/3.7/tutorial/>, sections 1 to 4.

1. Accept two int values from the user and return their product. If the product is greater than 1000, then return their sum.
2. Given a range of numbers. Iterate from i-th number to the end number and print the sum of the current number and previous number.
3. Given a list of ints, return True if first and last number of a list is same
4. Given a list of numbers, Iterate it and print only those numbers which are divisible of 5
5. Return the number of times that the string “Emma” appears anywhere in the given string: “Emma is a good developer. Emma is also a writer”
6. Given a two list of ints create a third list such that should contain only odd numbers from the first list and even numbers from the second list
7. Given 2 strings, s1 and s2, create a new string by appending s2 in the middle of s1
8. Given 2 strings, s1, and s2 return a new string made of the first, middle and last char each input string
9. Given a string input Count all lower case, upper case, digits, and special symbols
10. Find all occurrences of “USA” in given string ignoring the case
11. Given a string, return the sum and average of the digits that appear in the string, ignoring all other characters
12. Given an input string, count occurrences of all characters within a string

## Exercise Set 2

Data Structures - List, Set, Tuple, Dictionary.

Refer to <https://docs.python.org/3.7/tutorial/datastructures.html>

1. Given two list. Create a third list by picking an odd-index element from the first list and even index elements from second.

```
listOne = [3, 6, 9, 12, 15, 18, 21]
listTwo = [4, 8, 12, 16, 20, 24, 28]
```

2. Given an input list removes the element at index 4 and add it to the 2nd position and also, at the end of the list

```
sampleList = [34, 54, 67, 89, 11, 43, 94]
```

3. Given a list slice it into a 3 equal chunks and revert each list

```
sampleList = [11, 45, 8, 23, 14, 12, 78, 45, 89]
```

4. Given a list iterate it and count the occurrence of each element and create a dictionary to show the count of each element

```
sampleList = [11, 45, 8, 11, 23, 45, 23, 45, 89]
```

5. Given a two list of equal size create a set such that it shows the element from both lists in the pair

```
firstList = [2, 3, 4, 5, 6, 7, 8]
secondList = [4, 9, 16, 25, 36, 49, 64]
```

6. Given a following two sets find the intersection and remove those elements from the first set

```
firstSet = {23, 42, 65, 57, 78, 83, 29}
secondSet = {57, 83, 29, 67, 73, 43, 48}
```

7. Given two sets, Checks if One Set is Subset or superset of Another Set. if the subset is found delete all elements from that set

```
firstSet = {57, 83, 29}
secondSet = {57, 83, 29, 67, 73, 43, 48}
```

8. Iterate a given list and Check if a given element already exists in a dictionary as a key's value if not delete it from the list

```
rollNumber = [47, 64, 69, 37, 76, 83, 95, 97]
sampleDict = {'Jhon':47, 'Emma':69, 'Kelly':76, 'Jason':97}
```

9. Given a dictionary get all values from the dictionary and add it in a list but don't add duplicates

```
speed = {'jan':47, 'feb':52, 'march':47, 'April':44, 'May':52, 'June':53,  
'july':54, 'Aug':44, 'Sept':54}
```

10. Remove duplicate from a list and create a tuple and find the minimum and maximum number

```
sampleList = [87, 52, 44, 53, 54, 87, 52, 53]
```