# Python Basic Exercise

#### **Question 1:** Accept two int values from user and return their product. If the product is greater than 1000, then return their sum

**Question 2:** Given a range of numbers. Iterate from o^th number to the end number and print the sum of the current number and previous number

**Question 3:** Accept string from the user and display only those characters which are present at an even index

For example str = "pynative" so you should display ‘p’, ‘n’, ‘t’, ‘v’.

**Question 4:** Given a string and an int n, remove characters from string starting from zero upto n and return a new string

Note: n must be less than length of string. For example, removeChars("pynative", 4) so output must be “tive”.

**Question 5:** Given a list of ints, return True if first and last number of a list is same

**Question 6:** Given a list of numbers, Iterate it and print only those numbers which are divisible of 5

**Question 7:** Return the number of times that the string “Emma” appears anywhere in the given string

Given string is “Emma is a good developer. Emma is also a writer” and output should be 2.

**Question 8:** Print the following pattern

* **1**
* **2 2**
* **3 3 3**
* **4 4 4 4**
* **5 5 5 5 5**

#### **Question 9:** Reverse a given number and return true if it is the same as the original number

**Question 10:** 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

**Python String Exercise**

**Question 1:** Given a string of odd length greater 7, return a string made of the middle three chars of a given String

For example: –

* getMiddleThreeChars("JhonDipPeta") → "Dip"
* getMiddleThreeChars("Jasonay") → "son"

#### **Question 2:** Given 2 strings, s1 and s2, create a new string by appending s2 in the middle of s1

For example: –

* appendMiddle("Chrisdem", IamNewString) → "ChrIamNewStringisdem"

**Question 3:** Given 2 strings, s1, and s2 return a new string made of the first, middle and last char each input string

For example: – mixString("America", "Japan") = ""AJrpan"

**Question 4:** arrange String characters such that lowercase letters should come first

Given input String of combination of the lower and upper case arrange characters in such a way that all lowercase letters should come first.

For Example: –

* lowercaseFirst("PyNaTive") = aeiNPTvy

**Question 5:** Given an input string Count all lower case, upper case, digits, and special symbols

For example: –

* findDigitsCharsSymbols("P@#yn26at^&i5ve") = Chars = 8 Digits = 3 Symbol = 4

**Question 6:** Given two strings, s1 and s2, create a mix String

Note: create a third-string made of the first char of the last char of b, the second char of the second last char of b, and so on. Any leftover chars go at the end of the result.

For Example: –

* mixString("Pynative", "Website") = PeytniastbievWe

**Question 7:** String characters balance Test

#### We’ll say that a String s1 and s2 is balanced if all the chars in the string1 are there in s2. characters position doesn’t matter.

For Example: –

* stringBalanceCheck(yn, Pynative) = True

**Question 8:** Find all occurrences of “USA” in given string ignoring the case For Example: –

* countOccurrences("Welcome to USA. usa awesome, isn't it?") = 2

**Question 9:** Given a string, return the sum and average of the digits that appear in the string, ignoring all other characters

For Example: –

* sumAndAverage("English = 78 Science = 83 Math = 68 History = 65") = sum 294 Percentage **is** 73.5

**Question 10:** Given an input string, count occurrences of all characters within a string

For Example:

* count("pynativepynvepynative") = {'p': 3, 'y': 3, 'n': 3, 'a': 2, 't': 2, 'i': 2, 'v': 3, 'e': 3}

# Python Data Structure

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

### For Example:

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

Expected Outcome: [6, 12, 18, 4, 12, 20, 28]

**Question 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

For example: List = [54, 44, 27, 79, 91, 41]

Expected Outcome: [54, 44, 79, 27, 91, 41, 79].

**Question 3**: Given a list slice it into a 3 equal chunks and rever each list For Example: sampleList = [1, 2, 3, 4, 5, 6, 7, 8, 9]

Expected Outcome:

* **FirstList = [3, 2, 1]**
* **sampleList = [6, 5, 4]**
* sampleList = [9, 8, 7]

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

For example: list = [10, 20, 30, 10, 20, 40, 50]

Expected Outcome: dict = {10: 2, 20: 2, 30: 1, 40: 1, 50: 1}

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

**For example:**

* **firstList = [1, 2, 3, 4, 5]**
* **secondList = [10, 20, 30, 40, 50]**

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

## For Example:

* **firstSet = {10, 30, 40 , 60, 45}**
* **secondSet = {20, 50, 10 , 40, 55}**

**Expected Outcome: firstSet = {30, 60, 45}**

**Question 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

## For Example:

* **firstSet = {27, 43, 34}**
* **secondSet = {34, 93, 22, 27, 43, 53, 48}**

## Expected Outcome:

* **First set is sub set of second set**
* **firstSet = {}**

**Question 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

## Given:

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

## Expected Outcome:

* **after removing unwanted elemnts from list [47, 69, 76, 97]**

**Question 9:** Given a dictionary get all values from the dictionary and add it in a list but don’t add duplicates

## Given:

* **speed ={'jan':47, 'feb':52, 'march':47, 'April':44, 'May':52,**

'June':53, 'july':54, 'Aug':44, 'Sept':54}

**Expected Outcome: [47, 52, 44, 53, 54]**

**Question 10**: Remove duplicate from a list and create a tuple and find the minimum and maximum number

## For Example:

* **sampleList = [87, 45, 41, 65, 94, 41, 99, 94]**

## Expected Outcome:

* **unique items [87, 45, 41, 65, 99]**
* **tuple (87, 45, 41, 65, 99)**
* **min: 41**
* **max: 99**