**Python Exercise 1: Write a Python program to make a simple arithmetic calculator.**

**Problem Statement**

We need to write a Python program that accepts two numbers and an arithmetic operator from the user and performs the arithmetic operation on those two numbers.

**Python Exercise 2: Write a Python program to print the sum of first n numbers.**

**Problem Statement**

You need to write a Python program that accepts an N positive integer from the user and print the sum upto that N number For example, if the user input 4 the program should print 10 N = 4 1 + 2 + 3 + 4 = **10**.

**Python Exercise 3: Write a Python program to print all the prime numbers between two intervals**

**Problem Statement**You need to write a Python script that asks the user to enter an interval of positive numbers and print all the prime numbers of that interval.

**Python Exercise 4: RockyDon Problem**

**Problem Statement**

Given a positive integer number n. You need to write a Python script that iterates through 1 to n. Prints Rocky if the number is divisible by 3, print Don if the number is divisible by 5, Print “RockyDon” if the number is divisible by both 3 and 5, else it simply print the number.

**Example**

n = 10 1 2 Rocky 4 Don Rocky 7 8 Rocky Don

**Python Exercise 5: Remove first n Characters from a string**

**Problem Statement**

You have given a non-empty string, and a positive integer number **n**(less than the length of the string). You need to write a Python function that returns by removing the first n characters from the string.

**Example**

string = "Hi There! Welcome to UDMS" n = 10 Remove the first 10 characters from the string. **Output:**Welcome to UDMS

**Python Exercise 6: Write a Python program that accepts a list of 10 float numbers from the user.**

**Problem Statement**

You need to write a Python program that accepts 10 float numbers from the user and adds them to a list.

**Example: If user enters 10 20 30 40 50 60 70 80 90 100**

**Output:**[10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 70.0, 80.0, 90.0, 100.00]

**Python Exercise 7: Python program to write data of one file to another except for lines 3, 6, and 9.**

**Problem statement**

You have given a file data.txt, you need to write a Python program that reads the content from data.txt and writes its all content to new\_data.txt except the lines 3, 6, and 9.

|  |  |
| --- | --- |
| **data.txt** | **new\_data.txt** |
| Line 1 data Line 2 data Line 3 data Line 4 data Line 5 data Line 6 data Line 7 data Line 8 data Line 9 data Line 10 data | Line 1 data Line 2 data Line 4 data Line 5 data Line 7 data Line 8 data Line 10 data |

**Python Exercise 8: Python program to print n number of the Fibonacci sequence using recursion.**

**Problem Statement**

A Fibonacci sequence is a series of integers that start from 0 and 1, and every next number is decided by the sum of the previous two numbers. You need to write a Python code that asks the user to enter the value **n,**representing the length of the sequence. And a recursive function to print a Fibonacci sequence of n length.

**Example**

n = 10 **Output:**0 0 1 2 3 5 8 13 21 34

**Python Exercise 9: Python program to print the first non-repeated character from a string.**

**Problem Statement:**You have given a string and you need to find the first non-repeated characters.

**Example**

given\_string = welcome to techgeekbuzz.com website **Output**l

**Python Exercise 10: Python program to count the occurrence of each item from a list.**

**Problem Statement**

You have given a list with repeated items and you need to write a script that counts the occurrence of every list item.

**Example**

given\_list = [10, 20, 30, 10, 30, 20, 20, 20, 40, 50] **Output =**{10 : 2, 20 : 4, 30: 2, 40: 1, 50: 1 }

## Capital indexes

Write a function named capital\_indexes. The function takes a single parameter, which is a string. Your function should return a list of all the indexes in the string that have capital letters.

For example, calling capital\_indexes("HeLlO") should return the list [0, 2, 4].

## Middle letter

Write a function named mid that takes a string as its parameter. Your function should extract and return the middle letter. If there is no middle letter, your function should return the empty string.

For example, mid("abc") should return "b" and mid("aaaa") should return "".

## Online status

The aim of this challenge is, given a dictionary of people's online status, to count the number of people who are online.

For example, consider the following dictionary:

statuses = {

"Alice": "online",

"Bob": "offline",

"Eve": "online",

}

In this case, the number of people online is 2.

Write a function named online\_count that takes one parameter. The parameter is a dictionary that maps from strings of names to the string "online" or "offline", as seen above.

Your function should return the number of people who are online.

## Tic tac toe input

Here's the backstory for this challenge: imagine you're writing a tic-tac-toe game, where the board looks like this:

1: X | O | X

-----------

2: | |

-----------

3: O | |

A B C

The board is represented as a 2D list:

board = [

["X", "O", "X"],

[" ", " ", " "],

["O", " ", " "],

]

Imagine if your user enters "C1" and you need to see if there's an X or O in that cell on the board. To do so, you need to translate from the string "C1" to row 0 and column 2 so that you can check board[row][column].

Your task is to write a function that can translate from strings of length 2 to a tuple (row, column). Name your function get\_row\_col; it should take a single parameter which is a string of length 2 consisting of an uppercase letter and a digit.

For example, calling get\_row\_col("A3") should return the tuple (2, 0) because A3 corresponds to the row at index 2 and column at index 0in the board.

## Adding and removing dots

Write a function named add\_dots that takes a string and adds "." in between each letter. For example, calling add\_dots("test") should return the string "t.e.s.t".

Then, below the add\_dots function, write another function named remove\_dots that removes all dots from a string. For example, calling remove\_dots("t.e.s.t") should return "test".

If both functions are correct, calling remove\_dots(add\_dots(string)) should return back the original string for any string.

(You may assume that the input to add\_dots does not itself contain any dots.)