

# Title: Mastering Advanced Python: Tricky and Situational Programming

## Challenges Objective:

The objective of this assignment is to strengthen your problem-solving skills and enhance your understanding of core Python concepts such as strings, tuples, dictionaries, sets, lists, functions, lambda expressions, filters, and modules. This exercise aims to challenge your thinking, simulate real-world scenarios, and improve your ability to write efficient, modular, and optimized Python code.

### 1. Write a function to count the number of vowels and consonants in a given string.

```
In [ ]: def count_vowels_consonant():
    text = input("Enter the String: ").lower()
    vowels = 'aeiou'
    vowel_count = 0
    consonant_count = 0

    for char in text:
        if char.isalpha():
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1
    print("Number of Vowels: ", vowel_count)
    print("Number of Consonants: ", consonant_count)

count_vowels_consonant()
```

```
Enter the String: Ravi Kant
Number of Vowels: 3
Number of Consonants: 5
```

### 2. Write a function to return the minimum and maximum elements from a tuple.

```
In [ ]: #Function to find min and max in a tuple
def min_max(data):
    return min(data), max(data)
```

```
In [ ]: my_tuple = (10, 4, 7, 25, 1)
# call the function
minimum, maximum = min_max(my_tuple)
print("Minimum:", minimum)
print("Maximum:", maximum)
```

```
Minimum: 1
Maximum: 25
```

### 3. Write code to unpack the first two elements and the rest from a tuple of unknown length.

```
In [ ]: # suppose we have tuple
my_tuple = (10, 4, 7, 25, 1)
# unpack the first two elements and collecting the rest
a,b,*rest = my_tuple
# Displaying the result
print("First element:", a)
print("Second element:", b)
print("Rest of the elements:", rest)
```

First element: 10  
 Second element: 4  
 Rest of the elements: [7, 25, 1]

#### 4. Write a function that takes a string and returns the frequency of each word.

```
In [ ]: def word_frequency(text):
        words = text.lower().split()
        frequency = {}

        for word in words:
            frequency[word] = frequency.get(word,0) + 1
        return frequency
```

```
In [ ]: # taking input from the users
user_input = input("Enter a sentence:")

#call the function
result = word_frequency(user_input)
print("Word frequency:", result)
```

Enter a sentence:This is a test. This is a simple test  
 Word frequency: {'this': 2, 'is': 2, 'a': 2, 'test.': 1, 'simple': 1, 'test': 1}

#### 5. Write a function to find duplicate elements in a list.

```
In [ ]: def find_duplicate(lst):
        seen = set()
        duplicates = set()

        for item in lst:
            if item in seen:
                duplicates.add(item)
            else:
                seen.add(item)
        return list(duplicates)

In [ ]: my_list = [1, 2, 3, 2, 4, 5, 1, 6, 3, 3]
print("Duplicates:", find_duplicate(my_list))
```

Duplicates: [1, 2, 3]

#### 6. Use a filter with a lambda function to extract even numbers from a list.

```
In [ ]: # Sample list of numbers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
even_number = list(filter(lambda x: x % 2 == 0,numbers))
print("Even numbers:", even_number)
```

Even numbers: [2, 4, 6, 8, 10]

## 7. Create a Python module named mymath.py with functions for addition, subtraction, and multiplication. Import and use it.

```
In [ ]: # Create and save the module as a Python file
with open("mymath.py", "w") as f:
    f.write("""
def addition(a, b):
    return a + b

def subtraction(a, b):
    return a - b

def multiplication(a, b):
    return a * b
""")
```

```
In [ ]: import mymath

# Using the functions
x = 10
y = 5

print("Addition:", mymath.addition(x, y))
print("Subtraction:", mymath.subtraction(x, y))
print("Multiplication:", mymath.multiplication(x, y))
```

Addition: 15  
Subtraction: 5  
Multiplication: 50

## 8. Sort a List of Tuples Based on the Second Element

```
In [ ]: # sample list of tuple
tuple_list = [(1, 2), (4, 3), (2, 6), (5, 1)]

# Sort based on the second element (index 1 of each tuple)
sorted_list = sorted(tuple_list, key = lambda x:x[1])
print(sorted_list)
```

[(5, 1), (1, 2), (4, 3), (2, 6)]

## 9. Write a function to find the symmetric difference (elements not common) between two sets.

```
In [ ]: def symmetric_diff(set1,set2):
    return set1.symmetric_difference(set2)

#Sample set
a = {1, 2, 3, 4}
b = {3, 4, 5, 6}

result = symmetric_diff(a,b)
print("Symmetric Difference:",result)
```

Symmetric Difference: {1, 2, 5, 6}

**10. Write a function that returns a list of prime numbers up to a given number n.**

```
In [ ]: def prime_num(n):  
        primes = []  
        for num in range(2,n+1):  
            for i in range(2,num):  
                if num % i == 0:  
                    break  
            else:  
                primes.append(num)  
        return primes
```

```
In [ ]: print(prime_num(20))
```

[2, 3, 5, 7, 11, 13, 17, 19]

```
In [ ]:
```