

1. Write a Python function that accepts a string and calculates the number of upper case letters and lower case letters. Input string might contain special characters (say @, #). You need to ignore them while performing the calculation.

```
In [1]: my_string = 'AsDfGhJkLl23@'
```

```
In [2]: def case_count(string):
        fixed_string = ''.join([char for char in my_string if char.isalpha()])
        uppercase_count = 0
        lowercase_count = 0
        for char in fixed_string:
            if char.isupper():
                uppercase_count += 1
            else:
                lowercase_count += 1
        print(f'Number of uppercase letters = {uppercase_count}')
        print(f'Number of lowercase letters = {lowercase_count}')
```

```
In [3]: case_count(my_string)
```

```
Number of uppercase letters = 5
Number of lowercase letters = 4
```

2. Write a Python function that checks whether a number is in a given range (inclusive of high and low)

```
In [4]: def range_check(a,b,n):
        if a < b:
            if n >= a:
                if n <= b:
                    print(f'{n} is within the range {a}-{b}')
                else:
                    print(f'{n} is not within the range {a}-{b}')
            else:
                print(f'{n} is not within the range {a}-{b}')
        if a > b:
            if n <= a:
                if n >= b:
                    print(f'{n} is within the range {b}-{a}')
                else:
                    print(f'{n} is not within the range {b}-{a}')
            else:
                print(f'{n} is not within the range {b}-{a}')
```

```
In [5]: # Test 1
range_check(1,9,0)
range_check(1,9,1)
range_check(1,9,5)
range_check(1,9,9)
range_check(1,9,10)
```

```
0 is not within the range 1-9
1 is within the range 1-9
5 is within the range 1-9
9 is within the range 1-9
10 is not within the range 1-9
```

```
In [6]: # Test 2
range_check(9,1,0)
range_check(9,1,1)
range_check(9,1,5)
range_check(9,1,9)
range_check(9,1,10)
```

```
0 is not within the range 1-9
1 is within the range 1-9
5 is within the range 1-9
9 is within the range 1-9
10 is not within the range 1-9
```

3. Read a string and a pattern and return True if the pattern exists in the string otherwise False.

```
In [7]: # Scenario 1
my_string = 'This is a string'
pattern = 'is'
if pattern in my_string:
    print('True')
else:
    print('False')
```

```
True
```

```
In [8]: # Scenario 2
my_string = 'This is a string'
pattern = 'isa'
if pattern in my_string:
    print('True')
else:
    print('False')
```

```
False
```

```
In [9]: # Alternate solution - scenario 1
my_string = 'This is a string'
pattern = 'is'
if my_string.find(pattern) >= 0:
    print('True')
else:
    print('False')
```

```
True
```

```
In [10]: # Alternate solution - scenario 2
my_string = 'This is a string'
pattern = 'isa'
if my_string.find(pattern) >= 0:
    print('True')
else:
    print('False')
```

```
False
```

4. Write a Python program to find the factorial of a number

```
In [11]: def factorial(number):
        # number = int(input("Enter an integer:"))
        result = 1
        if number == 0:
            print("The factorial of 0 is 1")
        else:
            for i in range(1,number+1):
                result = result*i
            print(f'The factorial of {number} is {result}')
```

```
In [12]: factorial(6)
```

```
The factorial of 6 is 720
```

```
In [13]: factorial(0)
```

```
The factorial of 0 is 1
```

5. Implement a simple calculator : Read two numbers and the operation (addition/subtraction/multiplication/division/remainder/floor division), Output the corresponding result. [Hint: use if/elif/else]

```
In [14]: def calculator(number1,number2,operator):
        if operator.lower() == "addition":
            result = number1 + number2
        elif operator.lower() == "subtraction":
            result = number1 - number2
        elif operator.lower() == "multiplication":
            result = number1 * number2
        elif operator.lower() == "division":
            result = number1 / number2
        elif operator.lower() == "remainder":
            result = number1 % number2
        elif operator.lower() == "floor division":
            result = number1 // number2
        return result
```

```
In [15]: calculator(16,3,"addition")
```

```
Out[15]: 19
```

```
In [16]: calculator(16,3,"subtraction")
```

```
Out[16]: 13
```

```
In [17]: calculator(16,3,"multiplication")
```

```
Out[17]: 48
```

```
In [18]: calculator(16,3,"division")
```

```
Out[18]: 5.333333333333333
```

```
In [19]: calculator(16,3,"remainder")
```

```
Out[19]: 1
```

```
In [20]: calculator(16,3,"floor division")
```

```
Out[20]: 5
```

6. Write a Python program to calculate the area of a triangle

```
In [21]: def triangle_area(a,b,c):
        s = (a+b+c)/2
        area=(s*(s-a)*(s-b)*(s-c))**.5
        return area
```

```
In [22]: triangle_area(6,8,10)
```

```
Out[22]: 24.0
```

7. You are given three sides of a triangle: a, b and c. Return True if they constitute a rightangled triangle.

```
In [23]: def right_triangle_check(a,b,c):
        if (a**2 + b**2 == c**2) or (c**2 + b**2 == a**2) or (a**2 + c**2 == b**2):
            print("It is a right triangle!")
        else:
            print("It is not a right triangle!")
```

```
In [24]: right_triangle_check(3,4,5)
```

```
It is a right triangle!
```

```
In [25]: right_triangle_check(5,4,3)
```

```
It is a right triangle!
```

```
In [26]: right_triangle_check(3,5,4)
```

```
It is a right triangle!
```

```
In [27]: right_triangle_check(4,6,3)
```

```
It is not a right triangle!
```

8. Write a python program which returns the reverse of a number. Don't use string operations.

```
In [28]: def reverse_number():
        number = int(input("Enter an integer number: "))
        rev_number = 0

        while (number > 0):
            remainder = number % 10
            rev_number = (rev_number * 10) + remainder
            number = number // 10

        print(f'The reverse number is : {rev_number}')
```

```
In [29]: reverse_number()
```

```
Enter an integer number: 12345
The reverse number is : 54321
```

9. Take two strings from STDIN, returns True if one is the palindrome of another. Palindrome: a word, phrase, or sequence that reads the same backward as forward, e.g., madam

```
In [30]: def palindromes_check():
        string1 = input("Enter first string: ").lower()
        string2 = input("Enter second string: ").lower()

        if (string1 == string1[::-1]) & (string1[::-1] == string2):
            return True
        else:
            return False
```

```
In [31]: palindromes_check()
```

```
Enter first string: madam
Enter second string: Madam
```

```
Out[31]: True
```

10. Write a Python program to find the average of all the numbers in a given list.

```
In [32]: def list_average(my_list):
        average = sum(my_list) / len(my_list)
        return average
```

```
In [33]: num_list = [x for x in range(5)]
print(num_list)
print(f'Average of the list elements is = {list_average(num_list)}')
```

```
[0, 1, 2, 3, 4]
Average of the list elements is = 2.0
```

11. Write a Python program to combine values in a list of dictionaries.

```
In [34]: input_list = [{'item': 'item1', 'amount': 400},{'item': 'item2', 'amount': 300},\
({'item': 'item1', 'amount': 750}]
```

```
In [35]: def combined_output(item_list):
        combined_output = {}
        for item in item_list:
            if item['item'] not in combined_output:
                combined_output[item['item']] = item['amount']
            else:
                combined_output[item['item']] += item['amount']
        return combined_output
```

```
In [36]: combined_output(input_list)
```

```
Out[36]: {'item1': 1150, 'item2': 300}
```

```
In [37]: combined_output([])
```

```
Out[37]: {}
```

```
Empty list will return an empty dictionary.
```

```
In [38]: # Alternate solution
from collections import Counter

combined_values = Counter()
for item in input_list:
    combined_values[item['item']] += item['amount']
print(dict(combined_values))
```

```
{'item1': 1150, 'item2': 300}
```