Name-bhavvya mahawar
batch - 80
sapid -590029516

1. Write a program to count and display the number of capital letters in a given string.

```
def count_capital_letters(input_string):
    count = 0
    for char in input_string:
        if 'A' <= char <= 'Z':
            count += 1
    return count

# Example usage:
my_string = "Hello World! How Are You Today?"
capital_count = count_capital_letters(my_string)
print(f"The string: '{my_string}'")
print(f"Number of capital letters: {capital_count}")

my_string_2 = "PYTHON Programming Is FUN"
capital_count_2 = count_capital_letters(my_string_2)
print(f"\nThe string: '{my_string_2}'")
print(f"Number of capital letters: {capital_count_2}")
```

2. Count total number of vowels in a given string.

```
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    count = 0
    for char in input_string:
        if char in vowels:
            count += 1
    return count

# Example usage:
my_string_1 = "Programming is fun!"
vowel_count_1 = count_vowels(my_string_1)
print(f"The string: '{my_string_1}'")
print(f"Number of vowels: {vowel_count_1}")

my_string_2 = "HELLO world"
vowel_count_2 = count_vowels(my_string_2)
print(f"\nThe string: '{my_string_2}'")
print(f"Number of vowels: {vowel_count_2}")
```

Input a sentence and print words in separate lines.

```
sentence = input("Enter a sentence: ")
words = sentence.split()

print("\nWords in separate lines:")
for word in words:
    print(word)
```

4. WAP to enter a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left. Sample Input ABCDCDC CDC Sample Output 2

```
main_string = "ABCDCDC"
substring = "CDC"
occurrences = main_string.count(substring)

print(f"The main string: '{main_string}'")
print(f"The substring: '{substring}'")
print(f"The substring '{substring}' appears {occurrences} times in the string '{main_string}'.")
```

5. Given a string containing both upper and lower case alphabets. Write a Python program to count the number of occurrences of each alphabet (case insensitive) and display the same. Sample Input ABaBCbGc Sample Output 2A 3B 2C 1G

```
from collections import Counter

def count_alphabet_occurrences(input_string):

    processed_string = input_string.lower()


    alphabet_counts = Counter()

    for char in processed_string:

        if 'a' <= char <= 'z':
            alphabet_counts[char] += 1

    sorted_alphabets = sorted(alphabet_counts.keys())

    for char in sorted_alphabets:
        print(f"{alphabet_counts[char]}{char.upper()}")

# Sample Input
print("Sample Input:")
sample_input = "ABaBCbGc"
print(sample_input)
print("Sample Output:")
count_alphabet_occurrences(sample_input)

print("\nAnother example:")
count_alphabet_occurrences("Hello World")
```

6. Program to count number of unique words in a given sentence using sets.

```
import re

def count_unique_words(sentence):
    words = re.findall(r'\b\w+\b', sentence.lower())


    unique_words_set = set(words)

    return len(unique_words_set)

# Example usage:
my_sentence_1 = "This is a sample sentence. This sentence is a sample."
unique_count_1 = count_unique_words(my_sentence_1)
print(f"The sentence: '{my_sentence_1}'")
print(f"Number of unique words: {unique_count_1}")

my_sentence_2 = "Hello world, hello Python!"
unique_count_2 = count_unique_words(my_sentence_2)
print(f"\nThe sentence: '{my_sentence_2}'")
print(f"Number of unique words: {unique_count_2}")

my_sentence_3 = "Apple banana apple orange"
unique_count_3 = count_unique_words(my_sentence_3)
print(f"\nThe sentence: '{my_sentence_3}'")
print(f"Number of unique words: {unique_count_3}")
```

7. Create 2 sets s1 and s2 of n fruits each by taking input from user and find: a) Fruits which are in both sets s1 and s2

b) Fruits only in s1 but not in s2 c) Count of all fruits from s1 and s2

```
def get_fruit_set(set_name, num_fruits):
    fruits = set()
    print(f"\nEnter {num_fruits} fruits for {set_name}:")
    for i in range(num_fruits):
        fruit = input(f"Enter fruit {i+1}: ").strip().lower()
        fruits.add(fruit)
    return fruits

try:
    n = int(input("Enter the number of fruits for each set (n): "))
    if n <= 0:
        print("Number of fruits must be a positive integer.")
    else:
```

```
            s1 = get_fruit_set("set s1", n)

            s2 = get_fruit_set("set s2", n)

            print(f"\nSet s1: {s1}")
            print(f"Set s2: {s2}")
            common_fruits = s1.intersection(s2)
            print(f"\na) Fruits in both s1 and s2: {common_fruits}")
            only_in_s1 = s1.difference(s2)
            print(f"b) Fruits only in s1 but not in s2: {only_in_s1}")

            all_fruits_union = s1.union(s2)
            count_all_fruits = len(all_fruits_union)
            print(f"c) All unique fruits from s1 and s2: {all_fruits_union}")
            print(f"   Count of all unique fruits from s1 and s2: {count_all_fruits}")

    except ValueError:
        print("Invalid input. Please enter an integer for the number of fruits.")
```

8. Take two sets and apply various set operations on them : S1 = {Red ,yellow, orange , blue } S2 = {violet, blue , purple}

```
# Define the two sets
S1 = {"Red", "yellow", "orange", "blue"}
S2 = {"violet", "blue", "purple"}

print(f"Set S1: {S1}")
print(f"Set S2: {S2}")

union_set = S1.union(S2)
print(f"\nUnion of S1 and S2 (S1 | S2): {union_set}")
intersection_set = S1.intersection(S2)
print(f"Intersection of S1 and S2 (S1 & S2): {intersection_set}")

difference_s1_s2 = S1.difference(S2)
print(f"Elements in S1 but not in S2 (S1 - S2): {difference_s1_s2}")

difference_s2_s1 = S2.difference(S1)
print(f"Elements in S2 but not in S1 (S2 - S1): {difference_s2_s1}")

symmetric_difference_set = S1.symmetric_difference(S2)
print(f"Symmetric Difference of S1 and S2 (S1 ^ S2): {symmetric_difference_set}")

is_subset = S1.issubset(S2)
print(f"Is S1 a subset of S2? {is_subset}")

is_superset = S1.issuperset(S2)
print(f"Is S1 a superset of S2? {is_superset}")
```

Github repo link-https://github.com/mahawarbhavya/pythonexperiments.git