PCC-DS 391 Lab Assignment 3

1. Write a program that capitalizes the first letter of each word in a given sentence.

```
def capitalize_words(sentence):
    # Split the sentence into words
    words = sentence.split()
    # Capitalize the first letter of each word and join them back into a sentence
    capitalized_sentence = ' '.join(word.capitalize() for word in words)
    return capitalized_sentence
sentence = "hello world, this is a test sentence."
capitalized_sentence = capitalize_words(sentence)
print(capitalized_sentence)
```

Output:

```
Hello World, This Is A Test Sentence.
```

2. Write a program to reverse a given string.

```
# Define a sample string
sample_string = "Hello, World!"
# Reverse the string using slicing
reversed_string = sample_string[::-1]
# Print the result
print(f"The original string is: '{sample_string}'")
print(f"The reversed string is: '{reversed_string}'")
```

Output:

```
The original string is: 'Hello, World!'
The reversed string is: '!dlroW ,olleH'
```

3. Write a program to find all unique characters in a string by converting the string to a set.

```
# Define a sample string
sample_string = "Hello, World!"
# Convert the string to a set to find unique characters
unique_characters = set(sample_string)
# Print the result
print("The unique characters in the string are:")
print(unique_characters)
```

Output:

```
The unique characters in the string are:
{'e', 'H', 'r', '!', 'l', ' ', 'W', ',', 'o', 'd'}
```

4. Write a program to count all letter, digit and special symbol from a given string.

```
# Define a sample string
sample_string = "Hello, World! 123"
# Initialize counters
```

```
letter_count = 0
digit_count = 0
special_symbol_count = 0
# Iterate through each character in the string
for char in sample_string:
    if char.isalpha():
        letter_count += 1
    elif char.isdigit():
        digit_count += 1
    elif not char.isspace():
        special_symbol_count += 1
# Print the results
print(f"Total letters: {letter_count}")
print(f"Total digits: {digit_count}")
print(f"Total special symbols: {special_symbol_count}")
```

Output:

```
Total letters: 10
Total digits: 3
Total special symbols: 4
```

5. Write a program for remove of all character from a string except integer and returns a list of all the digits in the string.

```
# Define a sample string
sample_string = "Hello, World! 123"
# Initialize an empty string to store the result
digits_only = ""
# Initialize an empty list to store the digits
digits_list = []
# Iterate through each character in the string
for char in sample_string:
    if char.isdigit():
        digits_only += char
                            digits_list.append(char)
# Print the result
print(f"The string with only digits is: '{digits_only}'")
print(f"The list of digits in the string is: {digits_list}")
```

Output:

```
The string with only digits is: '123'
```

6. Write a program that counts the frequency of each character in a string and returns a dictionary with the characters as keys and their counts as values.

```
text = "hello world"
# Initialize an empty dictionary to store the frequency of each character
frequency_dict = {}
```

```
# Iterate through each character in the string
for char in text:
    # If the character is already in the dictionary, increment its count
    if char in frequency_dict:
        frequency_dict[char] += 1
    # If the character is not in the dictionary, add it with a count of 1
    else:
        frequency_dict[char] = 1
print(frequency_dict)
```

Output:

```
{'h': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 1, 'w': 1, 'r': 1, 'd': 1}
```

7. Write a program that replaces all vowels in a string with a specific character (e.g., '*').

```
# Define a sample string
sample_string = "Hello, World!"
# Define the specific character to replace vowels
replacement_char = '*'
# Define the vowels
vowels = "AEIOUaeiou"
# Replace vowels in the string
modified_string = ''.join([replacement_char if char in vowels else char for char in sample_string])
# Print the result
print(f"The modified string is: '{modified_string}'")
```

Output:

```
The modified string is: 'H*ll*, W*rld!'
```

8. Write a program to extract dates from a text. The dates should be in the format dd-mm-yyyy or dd/mm/yyyy.

```
import re
# Define a sample text containing dates
sample_text = "Today's date is 12-08-2024. Tomorrow's date will be 13/08/2024. The project started on 01-01-2023 and ended on 31/12/2023."
# Define the regular expression pattern for dates in dd-mm-yyyy or dd/mm/yyyy format
date_pattern = r'\b\d{2}[-/]\d{2}[-/]\d{4}\b'
# Use the findall method to extract all matching dates
dates = re.findall(date_pattern, sample_text)
# Print the extracted dates
print("Extracted dates:", dates)
```

Output:

```
Extracted dates: ['12-08-2024', '13/08/2024', '01-01-2023', '31/12/2023']
```

PCC-DS 391 Lab Assignment 3

9. Write a program to check if a given string is a valid phone number. Assume valid phone numbers are of the form 123-456-7890.

```
import re
def is_valid_phone_number(phone_number):
    # Regular expression pattern for the phone number format 123-456-7890
    pattern = r'^\d{3}-\d{3}-\d{4}$'
    # Check if the phone number matches the pattern
    if re.match(pattern, phone_number):
        return True
    else:
        return False
phone_number = input("Enter a phone number: ")
if is_valid_phone_number(phone_number):
    print(f"{phone_number} is a valid phone number.")
else:
    print(f"{phone_number} is not a valid phone number.")
```

Output:

```
Enter a phone number: 123-456-7890
123-456-7890 is a valid phone number.
```

10. Write a program that splits a string by multiple delimiters (e.g., commas, semicolons, and spaces).

```
import re
input_string = "apple, orange;banana  grape;pear"
# Define a regex pattern for the delimiters (commas, semicolons, and spaces)
pattern = r'[,\s;]+'
# Use re.split to split the string by the specified delimiters
split_result = re.split(pattern, input_string)
print("Split result:", split_result)
```

11. Write a program that removes all hashtags from a given text.

```
import re
text = "I love #Python and #coding. #HappyCoding"
# Regular expression pattern to match hashtags
hashtag_pattern = r'#\w+'
# Substitute hashtags with an empty string
cleaned_text = re.sub(hashtag_pattern, '', text)
# Remove any extra spaces left after removing hashtags
cleaned_text = re.sub(r'\s+', ' ', cleaned_text).strip()
print(cleaned_text)
```

Output:

```
I love and .
```