Day 2 Python Programming

1) Right Angle Triangle

def right\_angle\_triangle(rows):

for i in range(1, rows + 1):

print('\*' \* i)

num\_rows = 5

right\_angle\_triangle(num\_rows)

2)Left angle triangle

def left\_angle\_triangle(rows):

for i in range(rows, 0, -1):

print(' ' \* (rows - i) + '\*' \* i)

num\_rows = 5

left\_angle\_triangle(num\_rows)

3)Removing Duplicate elements from the list

def remove\_duplicates(lst):

return list(set(lst))

original\_list = [1, 2, 2, 3, 4, 4, 5]

unique\_list = remove\_duplicates(original\_list)

print(f"Original list: {original\_list}")

print(f"List after removing duplicates: {unique\_list}")

4)Duplicate elements in the list and its count

from collections import Counter

def find\_duplicates(lst):

counts = Counter(lst)

duplicates = {item: count for item, count in counts.items() if count > 1}

return duplicates

original\_list = [1, 2, 2, 3, 4, 4, 5, 5, 5]

duplicates = find\_duplicates(original\_list)

print(f"Original list: {original\_list}")

print(f"Duplicates and their counts: {duplicates}")

5)Character frequencies

from collections import Counter

def character\_frequencies(s):

return dict(Counter(s))

input\_string = "hello world"

frequencies = character\_frequencies(input\_string)

print(f"Input string: {input\_string}")

print(f"Character frequencies: {frequencies}")

6)Reversed string

def reverse\_string(s):

return s[::-1]

input\_string = "hello"

reversed\_string = reverse\_string(input\_string)

print(f"Original string: {input\_string}")

print(f"Reversed string: {reversed\_string}")

7)Concatenated String

def concatenate\_strings(s1, s2):

return s1 + s2

string1 = "hello"

string2 = "world"

concatenated\_string = concatenate\_strings(string1, string2)

print(f"String 1: {string1}")

print(f"String 2: {string2}")

print(f"Concatenated string: {concatenated\_string}")

8) Pascal Triangle

def generate\_pascals\_triangle(rows):

triangle = []

for i in range(rows):

row = [1] \* (i + 1)

for j in range(1, i):

row[j] = triangle[i-1][j-1] + triangle[i-1][j]

triangle.append(row)

return triangle

def print\_pascals\_triangle(triangle):

for row in triangle:

print(' '.join(map(str, row)).center(2 \* len(triangle)))

num\_rows = 5

pascals\_triangle = generate\_pascals\_triangle(num\_rows)

print(f"Pascal's Triangle with {num\_rows} rows:")

print\_pascals\_triangle(pascals\_triangle)

9)Concatenate without + operator

def concatenate\_strings(s1, s2):

# Using the join method to concatenate

return ''.join([s1, s2])

string1 = "hello"

string2 = "world"

concatenated\_string = concatenate\_strings(string1, string2)

print(f"String 1: {string1}")

print(f"String 2: {string2}")

print(f"Concatenated string: {concatenated\_string}")

10)Vowels and consonant in the string

def count\_vowels\_and\_consonants(s):

vowels = 'aeiouAEIOU'

vowel\_count = 0

consonant\_count = 0

for char in s:

if char.isalpha(): # Check if the character is a letter

if char in vowels:

vowel\_count += 1

else:

consonant\_count += 1

return vowel\_count, consonant\_count

input\_string = "Hello World!"

vowels\_count, consonants\_count = count\_vowels\_and\_consonants(input\_string)

print(f"Input string: {input\_string}")

print(f"Number of vowels: {vowels\_count}")

print(f"Number of consonants: {consonants\_count}")