DAY—2

1.RIGHT TRIANGLE

# Number of rows

n = 5

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

print('\*' \* i)

2.LEFT TRIANGLE

n = 5

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

print('\*' \* i)

3.PASCAL TRIANGLE

def print\_pascals\_triangle(n):

triangle = [[1]]

for i in range(1, n):

row = [1]

for j in range(1, i):

row.append(triangle[i-1][j-1] + triangle[i-1][j])

row.append(1)

triangle.append(row)

for row in triangle:

print(' '.join(map(str, row)))

print\_pascals\_triangle(5)

4.REMOVE DUPLICATE

def remove\_duplicates(lst):

return list(dict.fromkeys(lst))

lst = [1, 2, 2, 3, 4, 4, 5]

print(remove\_duplicates(lst))

5.PRINT DUPLICATE

From collections import counter

a=[1,2,2,3,4,5,6]

frequency=counter(a)

print(frequency)

6.FREQUENCY OF CHARACTERS

From collections import counter

String=”saveetha”

Frequency=counter(string)

Print(frequency)

7.REVERSE(SLICING)

original\_string = "Hello, world!"

reversed\_string = original\_string[::-1]

print(reversed\_string)

8.EMAIL IS VALID OR NOT

from email\_validator import validate\_email, EmailNotValidError

def is\_valid\_email(email):

try:

# Validate the email

v = validate\_email(email)

# Update with the normalized form

email = v["email"]

return True

except EmailNotValidError as e:

# Email is not valid, exception message is human-readable

print(str(e))

return False

# Example usage

email = "example@example.com"

if is\_valid\_email(email):

print(f"'{email}' is a valid email address.")

else:

print(f"'{email}' is not a valid email address.")

9.CONCATE WITHOUT + OPERATOR

def concatenate\_strings(s1, s2):

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

str1 = "Hello"

str2 = "World"

result = concatenate\_strings(str1, str2)

print(result)

10.VOWELS AND CONSONANTS

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

vowels = "aeiouAEIOU"

count\_v = sum(1 for char in s if char in vowels)

count\_c = sum(1 for char in s if char.isalpha() and char not in vowels)

print(f"Vowels: {count\_v}")

print(f"Consonants: {count\_c}")

s = "Hello World"