Day 4 lab python programs

1.def fizz\_buzz(n):

result = []

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

if i % 3 == 0 and i % 5 == 0:

result.append("FizzBuzz")

elif i % 3 == 0:

result.append("Fizz")

elif i % 5 == 0:

result.append("Buzz")

else:

result.append(str(i))

return result

# Test cases

test\_cases = [3, 5, 15, 10, 20]

for n in test\_cases:

print("Output:", fizz\_buzz(n))

2.

def calculate\_users(total\_users, staff\_users):

student\_users = total\_users - staff\_users

non\_teaching\_staff\_users = staff\_users // 3

return student\_users, non\_teaching\_staff\_users

def main():

try:

total\_users = int(input("Total Users: "))

staff\_users = int(input("Staff Users: "))

if total\_users < 0 or staff\_users < 0:

print("Invalid input. Users cannot be negative.")

return

student\_users, non\_teaching\_staff\_users = calculate\_users(total\_users, staff\_users)

print("Student Users:", student\_users)

print("Non-teaching Staff Users:", non\_teaching\_staff\_users)

except ValueError:

print("Invalid input. Please enter valid integer values.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

3.

def smallerNumbersThanCurrent(nums):

result = []

for i in range(len(nums)):

count = 0

for j in range(len(nums)):

if i != j and nums[j] < nums[i]:

count += 1

result.append(count)

return result

# Test cases

test\_cases = [

[8, 1, 2, 2, 3],

[6, 5, 4, 8],

[7, 7, 7, 7],

[1, 2, 3, 5, 5, 6],

[0, 0, 0, 0]

]

for nums in test\_cases:

output = smallerNumbersThanCurrent(nums)

print("Input:", nums)

print("Output:", output)

print()

4.

def isPalindrome(s):

# Convert the string to lowercase and remove non-alphanumeric characters

s = ''.join(c for c in s if c.isalnum()).lower()

# Check if the modified string is equal to its reverse

return s == s[::-1]

# Test cases

test\_cases = [

"A man, a plan, a canal: Panama",

"race a car",

" ",

"madam",

"honest"

]

for s in test\_cases:

output = isPalindrome(s)

print("Input:", s)

print("Output:", output)

print()

5.

def minJumps(arr):

n = len(arr)

if n <= 1:

return 0

# Initialize an array to store minimum jumps for each index

jumps = [float('inf')] \* n

jumps[0] = 0

for i in range(1, n):

for j in range(i):

if j + arr[j] >= i:

jumps[i] = min(jumps[i], jumps[j] + 1)

break

return jumps[-1] if jumps[-1] != float('inf') else -1

# Test cases

test\_cases = [

[1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9],

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],

[2, 3, 1, 1, 4],

[1, 3, 6, 1, 0, 9],

[2, 3, 0, 1, 4]

]

for arr in test\_cases:

output = minJumps(arr)

print("Input:", arr)

print("Output:", output)

print()

6.

def delchar(s, c):

if len(c) != 1:

return s

new\_string = ''.join([char for char in s if char != c])

return new\_string

# Test cases

test\_cases = [

("Good evening", "e"),

("Take care", "a"),

("123456s", "1"),

("Red rose", "o"),

("Flower", "z")

]

for s, c in test\_cases:

output = delchar(s, c)

print("Input string:", s)

print("Character to be deleted:", c)

print("String after character is removed:", output)

print()

7.

def countSortedVowelStrings(n):

# Initialize a 2D DP array to store the counts

dp = [[0] \* 5 for \_ in range(n + 1)]

# Initialize base cases

for i in range(5):

dp[1][i] = 1

# Fill up the DP array

for i in range(2, n + 1):

for j in range(5):

dp[i][j] = dp[i][j - 1] + dp[i - 1][j]

# The total count is the sum of counts for strings of length n

return sum(dp[n])

# Test cases

test\_cases = [1, 2, 33, 55, 32]

for n in test\_cases:

output = countSortedVowelStrings(n)

print("Input:", n)

print("Output:", output)

print()

8.

def romanToInt(s):

roman\_to\_int = {

"I": 1,

"V": 5,

"X": 10,

"L": 50,

"C": 100,

"D": 500,

"M": 1000

}

result = 0

prev\_value = 0

for char in reversed(s):

value = roman\_to\_int[char]

if value >= prev\_value:

result += value

else:

result -= value

prev\_value = value

return result

# Test cases

test\_cases = ["III", "LVIII", "MCMXCIV", "LV", "MMl"]

for s in test\_cases:

output = romanToInt(s)

print("Input:", s)

print("Output:", output)

print()

9.

def get\_season(month, day):

if (month == "March" and day >= 20) or (month == "April" or month == "May") or (month == "June" and day < 21):

return "spring"

elif (month == "June" and day >= 21) or (month == "July" or month == "August") or (month == "September" and day < 22):

return "summer"

elif (month == "September" and day >= 22) or (month == "October" or month == "November") or (month == "December" and day < 21):

return "fall"

else:

return "winter"

def main():

month = input("Enter the month: ").capitalize()

day = int(input("Enter the date: "))

season = get\_season(month, day)

print("The season is currently", season)

# Test cases

main()

10.

def isScramble(s1, s2, memo):

if len(s1) != len(s2):

return False

if len(s1) == 1:

return s1 == s2

if (s1, s2) in memo:

return memo[(s1, s2)]

n = len(s1)

for i in range(1, n):

if (isScramble(s1[:i], s2[:i], memo) and isScramble(s1[i:], s2[i:], memo)) or \

(isScramble(s1[:i], s2[n-i:], memo) and isScramble(s1[i:], s2[:n-i], memo)):

memo[(s1, s2)] = True

return True

memo[(s1, s2)] = False

return False

def scrambleString(s1, s2):

memo = {}

return isScramble(s1, s2, memo)

# Test cases

test\_cases = [

("great", "rgeat"),

("abcde", "caebd"),

("a", "a"),

("ab", "ad"),

("10", "-5")

]

for s1, s2 in test\_cases:

output = scrambleString(s1, s2)

print("Input s1:", s1)

print("Input s2:", s2)

print("Output:", output)

print()