**NAME: Suhaib**

**Roll No: 2K24-CSE-145**

**Subject: Data Structures.**

**Course: BSCS PE**

**Submitted To: Dr. Gulsher Laghari.**

DRIVE LINK: https://docs.google.com/document/d/17-t2RaP28ls36lD6\_4R7TJzuD6V72EON/edit?usp=drive\_link&ouid=114591842700172387790&rtpof=true&sd=true

Q.1 Write a program to reverse an array using stack data structure.

**PROGRAM:**

# Reverse an array using stack in Python

def reverse\_array(arr):

stack = [] # create a stack

# Push all elements of array into stack

for item in arr:

stack.append(item)

# Pop elements from stack back into array

for i in range(len(arr)):

arr[i] = stack.pop()

return arr

# Example usage:

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

print("Original array :", arr)

reversed\_arr = reverse\_array(arr)

print("Reversed array :", reversed\_arr)

**OUTPUT:**

Original array : [1, 2, 3, 4, 5]

Reversed array : [5, 4, 3, 2, 1]

Q.2 Write a program to match the parentheses stored in a string using stack data structure.

**PROGRAM:**

# Check if parentheses are matching using stack in Python

def matching\_parentheses(s):

stack = []

matching\_bracket = {')': '(', '}': '{', ']': '['}

for char in s:

if char in "({[":

stack.append(char)

elif char in ")}]":

if not stack or stack.pop() != matching\_bracket[char]:

return False

return len(stack) == 0

# Example usage:

string1 = "([{}])"

string2 = "([{]})"

print(f"String: {string1} -> Matching: {matching\_parentheses(string1)}")

print(f"String: {string2} -> Matching: {matching\_parentheses(string2)}")

**OUTPUT:**

String: ([{}]) -> Matching: True

String: ([{]}) -> Matching: False

Q.3 Write a program to calculate the sum of all integer elements in an integer array by implementing a recursive sum method/ function.

**PROGRAM:**

# Calculate sum of array elements using recursion in Python

def recursive\_sum(arr, n):

if n == 0:

return 0

else:

return arr[n-1] + recursive\_sum(arr, n-1)

# Example usage:

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

print("Array elements :", arr)

total = recursive\_sum(arr, len(arr))

print("Sum of array elements :", total)

**OUTPUT:**

Array elements : [1, 2, 3, 4, 5]

Sum of array elements : 15