**MCQs**

Q1. Ans - A) "Invalid input"

Q2. Ans - A) "Error: Not a positive integer"

Q3. Ans - A) "Result: Infinity"

Q4. Ans - B) [1, 4, 9, 16, 25]

Q5. Ans - A) 2

Q6. Ans - A) {1, 2, 3, 5, 6, 7, 9, 10}

Q7. Ans - B) 20

Q8. Ans -A) !dlroW ,olleH

Q9. Ans - B) 3

Q10. Ans - D) ValueError

Q11. Ans - B) 15

Q12. Ans - B) TypeError

Q13. Ans - A) 5.0 Done

Q14. Ans - B) 5

Q15. Ans - A) [1,2,3,4]

Q16. Ans - B) [2,4]

Q17. Ans - A) Hello Alice

Q18. Ans - B) 11

Q19. Ans - C) 120

Q20. Ans - B) 3

Q21. Ans - A) True

Q22. Ans - B) “HELLO, ALICE”

Q23. Ans - A) “Time taken: 2.0 seconds”

Q24. Ans - A) “Arguments: 3, 4, Result: 7”

Q25. Ans - B) 55

Q26. Ans - D) admin panel accessed. Access Denied

Q27. Ans - D) “Max Attempts reached”

**PROGRAMMING**

Q5. Circular picnic

def maxContiSum(arr):

sum = 0

for i in arr:

sum += i

print(sum)

arr = [10, -4, 1, 3, 3]

maxContiSum(arr)

Q6. Maximum Subarray Sum

def maxSum(a, size):

newMax = -1

max= 0

for i in range(0, size):

max = max + a[i]

if (newMax < max):

newMax = max

if max < 0:

max = 0

print(newMax)

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

maxSum(arr, len(arr))

Q7. Longest Common Subsequence

def lonComSeq(a, b, m, n):

if m == 0 or n == 0:

return 0;

elif a[m-1] == b[n-1]:

return 1 + lonComSeq(a, b, m-1, n-1);

else:

return max(lonComSeq(a, b, m, n-1), lonComSeq(a, b, m-1, n));

text1 = "AGGTAB"

text2 = "GXTXAYB"

print(lonComSeq(text1, text2, len(text1), len(text2)))

Q8. Matrix Spiral Order

def spiralPrint(m, n, a):

k = 0

l = 0

''' k - starting row index

m - ending row index

l - starting column index

n - ending column index '''

while (k < m and l < n):

for i in range(l, n):

print(a[k][i], end=" ")

k += 1

for i in range(k, m):

print(a[i][n - 1], end=" ")

n -= 1

if (k < m):

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

print(a[m - 1][i], end=" ")

m -= 1

if (l < n):

for i in range(m - 1, k - 1, -1):

print(a[i][l], end=" ")

l += 1

# Driver Code

a = [[1,2,3],

[4,5,6],

[7,8,9]]

row = 3

col = 3

spiralPrint(row, col, a)

Q1. Employee Performance Evalution

def checkPerformance(emp):

per = []

for i in emp:

a = {}

a["name"] = i["name"]

a["performance\_score"] = i["scores"]['Quality of Work']\*i["weights"]["Quality of Work"] + i["scores"]["Team Collaboration"]\*i["weights"]["Team Collaboration"]

per.append(a)

print(per)

employees = [

{"name": "John", "scores": {"Quality of Work": 90, "Team Collaboration": 80}, "weights": {"Quality of Work": 0.6, "Team Collaboration": 0.4}},

{"name": "Alice", "scores": {"Quality of Work": 85, "Team Collaboration": 95}, "weights": {"Quality of Work": 0.5, "Team Collaboration": 0.5}}

]

checkPerformance(employees)