**ASSIGNMENT – 1**

**DATA SCIENCE & GEN AI LLMS**

H NO - 2406DGAL135

CHINNAM SHIVA

 **Question 1:**

**Number game between user and computer. The user starts by entering either 1 or 2 or 3 digits starting from 1 sequentially. The computer can return either 1 or 2 or 3 next digits in sequence, starting from the max number played by the user. User enters the next 1 or 2 or 3 next digits in sequence, starting from the max number played by the computer. Whoever reaches 20 first wins the game.**

**Note:**

**- the numbers should be in sequence starting from 1.**

**- minimum number user or computer should pick is at least 1 digit in sequence**

**- maximum number user or computer can pick only 3 digits in sequence**

**Program:-**

def number\_game():

 current\_number = 0

 while current\_number < 20:

 player\_input = input("Enter 1 to 3 sequential numbers (e.g., '1 2 3'): ")

 player\_numbers = list(map(int, player\_input.split()))

 if len(player\_numbers) < 1 or len(player\_numbers) > 3 or any(num != current\_number + i + 1 for i, num in enumerate(player\_numbers)):

 print("Invalid input. Make sure to enter 1 to 3 sequential numbers starting from", current\_number + 1)

 continue

 current\_number += len(player\_numbers)

 print(f"player's turn: {', '.join(map(str, player\_numbers))} -> Current total: {current\_number}")

 if current\_number >= 20:

 print("player wins!")

 break

 computer\_numbers = []

 for \_ in range(min(3, 20 - current\_number)):

 computer\_numbers.append(current\_number + 1)

 current\_number += 1

 if current\_number >= 20:

 break

 print(f"Computer's turn: {', '.join(map(str, computer\_numbers))} -> Current total: {current\_number}")

 if current\_number >= 20:

 print("Computer wins!")

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

 number\_game()

**output :-**

Enter 1 to 3 sequential numbers (e.g., '1 2 3'): 1

player's turn: 1 -> Current total: 1

Computer's turn: 2, 3, 4 -> Current total: 4

Enter 1 to 3 sequential numbers (e.g., '1 2 3'): 5 6

player's turn: 5, 6 -> Current total: 6

Computer's turn: 7, 8, 9 -> Current total: 9

Enter 1 to 3 sequential numbers (e.g., '1 2 3'): 10 11

player's turn: 10, 11 -> Current total: 11

Computer's turn: 12, 13, 14 -> Current total: 14

Enter 1 to 3 sequential numbers (e.g., '1 2 3'): 15

player's turn: 15 -> Current total: 15

Computer's turn: 16, 17, 18 -> Current total: 18

Enter 1 to 3 sequential numbers (e.g., '1 2 3'): 19 20

player's turn: 19, 20 -> Current total: 20

player wins!

=== Code Execution Successful ===

**Question 2:**

**Develop a function called ncr(n,r) which computes r-combinations of n-distinct object . use this function to print pascal triangle, where number of rows is the input**

**Program:**

def fact(n):

 i=1

 f=1

 while (i<=n):

 f=f\*i

 i+=1

 return f

def ncr(n,r):

 return fact(n)//(fact(n-r)\*fact(r))

def triangle(r):

 for i in range(r):

 print(' ' \* (r - i), end='')

 for j in range(i+1):

 print(ncr(i,j), end=' ')

 print()

n= int(input("Enter n rows : "))

triangle(n)

**output:-**

Enter n rows : 8

 1

 1 1

 1 2 1

 1 3 3 1

 1 4 6 4 1

 1 5 10 10 5 1

 1 6 15 20 15 6 1

 1 7 21 35 35 21 7 1

=== Code Execution Successful ===

**Question 3:**

**Read a list of n numbers during runtime. Write a Python program to print the repeated elements with frequency count in a list**.

**Program :**

list1 = list(map(int,input("Enter numbers : ").split()))

count = {}

for i in list1:

 if i in count:

 count[i] += 1

 else:

 count[i] = 1

print(count)

for i , j in count.items():

 if j>1:

 print('Element' ,i ,'has come', j ,'times')

**output :**

Enter numbers : 2 1 2 3 4 5 1 3 6 2 3 4

{2: 3, 1: 2, 3: 3, 4: 2, 5: 1, 6: 1}

Element 2 has come 3 times

Element 1 has come 2 times

Element 3 has come 3 times

Element 4 has come 2 times

=== Code Execution Successful ===

**Question 4:-**

**Develop a python code to read matric A of order 2X2 and Matrix B of order 2X2 from a file and perform the addition of Matrices A & B and Print the results.**

**Program :-**

def read\_matrix(matrices.txt, matrix\_num):

 matrix = []

 with open(matrices.txt, 'r') as file:

 lines = file.readlines()

 start\_index = lines.index(f'Matrix{matrix\_num}:\n') + 1

 for i in range(start\_index, start\_index + 2):

 row = list(map(int, lines[i].split()))

 matrix.append(row)

 return matrix

def add\_matrices(matrix1, matrix2):

 result = [

 [matrix1[i][j] + matrix2[i][j] for j in range(2)]

 for i in range(2)

 ]

 return result

matrix1 = read\_matrix('matrices.txt', 1)

matrix2 = read\_matrix('matrices.txt', 2)

# Add matrices

result = add\_matrices(matrix1, matrix2)

print("Resultant Matrix:")

for row in result:

 print(row)

**input from file matrices.txt :-**

Matrix1:-

1 2

3 4

Matrix2:-

5 6

7 8

**output :-**

Resultant matrix:

[6, 8]

[10, 12]

**Question 5:-**

**Write a program that overloads the + operator so that it can add two objects of the class Fraction. Fraction can be considered of the for P/Q where P is the numerator and Q is the denominator**

**Program :-**

class Addition:

 def \_\_init\_\_(self, p, q):

 self.p = p

 self.q = q

 def display(self):

 print(self.p, "/", self.q)

 def \_\_add\_\_(self, other):

 fun1 = self.p \* other.q + other.p \* self.q

 fun2 = self.q \* other.q

 return f"{fun1}/{fun2}"

ob1 = Addition(3, 2)

ob2 = Addition(4, 1)

result = ob1 + ob2

print(result)

**output :-**

 11/2

=== Code Execution Successful ===