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

Example 1:

Player: 1 2

Computer played: [3, 4]

Player: 5 6 7

Computer played: [8, 9]

Player: 10

Computer played: [11, 12, 13]

Player: 14 15

Computer played: [16, 17, 18]

Player: 19 20

Player Wins!!!

Example 2:

Player: 1

Computer played: [2, 3]

Player: 4 5

Computer played: [6, 7, 8]

Player: 9 10

Computer played: [11]

Player: 12

Computer played: [13]

Player: 14 15

Computer played: [16]

Player: 17 18

Computer played: [19, 20]

Computer Wins!!!

Answer 1:

```
import random

def ComputersTurn(LastNum):
    # Computer picks 1 to 3 numbers in sequence
    turn = random.randint(1, 3)
    return list(range(LastNum + 1, LastNum + 1 + turn))

def PlayersTurn(LastNum):
    while True:
        try:
            PlayersInput = input("Enter 1 or 2 or 3 numbers in sequence separated by space: ").split()
            PlayersNum = [int(num) for num in PlayersInput]
            if len(PlayersNum) < 1 or len(PlayersNum) > 3:
                print("Enter either 1 or 2 or 3 numbers only")
                continue
            if PlayersNum[0] != LastNum + 1 or PlayersNum != list(range(PlayersNum[0], PlayersNum[0] +
len(PlayersNum))):
                print("Numbers must be in sequence starting from the next number and separated by space")
                continue
            return PlayersNum
```

```
        except ValueError:
            print("Enter numbers in sequence separated by space please")


def NumGame():
    LastNum = 0
    print("The number game begins now get ready to win!")
    while LastNum < 20:
        PlayersNum = PlayersTurn(LastNum)
        LastNum = PlayersNum[-1]
        print(f"Player: {PlayersNum}")
        if LastNum >= 20:
            print("Congragulations you Won :-D")
            break

        CompNum = ComputersTurn(LastNum)
        LastNum = CompNum[-1]
        print(f"Computer played: {CompNum}")
        if LastNum >= 20:
            print("Computer Wins :-( try again :-)")
            break

NumGame()
```

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.

Answer 2:

```
def CalcFact(n):
```

```
    if n == 0 or n == 1:
        return 1
    else:
        return n * CalcFact(n - 1)

def ncr(n, r):
    return CalcFact(n) // (CalcFact(r) * CalcFact(n - r))

def PascalTriangle(TriSize):
    for i in range(TriSize):
        print(' ' * (TriSize - i - 1), end='')
        for j in range(i + 1):
            print(ncr(i, j), end=' ')
        print()

TriSize = int(input("Enter the number of rows for Pascal's Triangle: "))
PascalTriangle(TriSize)
```

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.

Example :

Input:- [2,1,2,3,4,5,1,3,6,2,3,4]

Output:-

Element 2 has come 3 times

Element 1 has come 2 times

Element 3 has come 2 times

Element 4 has come 2 times

Element 1 has come 1 times

Element 6 has come 1 times

Answer 3:

```
def TimesRep(InputNum):
    times = {}
    for number in InputNum:
        if number in times:
            times[number] += 1
        else:
            times[number] = 1
    return times

def OutputTimes(times):
    for number, count in times.items():
        print(f"Element {number} has come {count} times")

InputStr = input("Enter the numbers separated by commas: ")
InputNum = [int(num.strip()) for num in InputStr.split(',')]
times = TimesRep(InputNum)
OutputTimes(times)
```

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.

Answer 4:

Text file content:

Matrix A:

1 2

3 4

Matrix B:

5 6

7 8

```
def ReadMatrix(FileName, MatrixName):
    matrix = []
    with open(FileName, 'r') as f:
        lines = f.readlines()
        start = lines.index(MatrixName + ':\n') + 1
        for i in range(start, start + 2):
            row = list(map(int, lines[i].strip().split()))
            matrix.append(row)
    return matrix

def AddMatrix(matrixA, matrixB):
    result = []
    for i in range(2):
        row = []
        for j in range(2):
            row.append(matrixA[i][j] + matrixB[i][j])
        result.append(row)
    return result

def OutputMatrix(matrix, OutPutMsg):
    for row in matrix:
        print(' '.join(str(num) for num in row))
```

```
FileName = 'InputFileForQuestion4.txt'
MatrixA = ReadMatrix(FileName, 'Matrix A')
MatrixB = ReadMatrix(FileName, 'Matrix B')
OutputMatrix(AddMatrix(MatrixA, MatrixB),"Matrix addition Output is:")
```

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.

Answer 5:

```
class Fraction:
    def __init__(self, NumeratorP, DenominatorQ):
        self.NumeratorP = NumeratorP
        self.DenominatorQ = DenominatorQ

    def __add__(self, other):
        if isinstance(other, Fraction):
            ResP = self.NumeratorP * other.DenominatorQ + other.NumeratorP * self.DenominatorQ
            ResQ = self.DenominatorQ * other.DenominatorQ
            return Fraction(ResP, ResQ)
        else:
            raise TypeError("Not a fraction")

    def __str__(self):
        return f"{self.NumeratorP}/{self.DenominatorQ}"

def InputFactions(prompt):
    NumeratorP = int(input(f"Enter the numerator (P) for {prompt}: "))
```

```
    DenominatorQ = int(input(f"Enter the denominator (Q) for {prompt}: "))
    return Fraction(NumeratorP, DenominatorQ)

print("Enter the first fraction:")
FirstFraction = InputFactions("Fraction 1")

print("Enter the second fraction:")
SecondFraction = InputFactions("Fraction 2")

Res = FirstFraction + SecondFraction

print(f"The result of adding {FirstFraction} and {SecondFraction} is {Res}")
```
