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!!!
```

Source code:

```
import random


def player_turn(current_max):
    while True:
        try:
            player_input = input(f"Player, enter your move (1-3 digits starting from {current_max + 1}): ")
            numbers = list(map(int, player_input.split()))


            if not all(x == current_max + 1 + i for i, x in enumerate(numbers)):
                print("Invalid move. Please enter sequential numbers starting from", current_max + 1)
                continue

            if len(numbers) < 1 or len(numbers) > 3:
                print("You must enter 1 to 3 numbers.")
                continue

            return numbers
        except ValueError:
            print("Please enter valid numbers.")
```

```
def computer_turn(current_max):
    count = random.randint(1, 3)
    numbers = list(range(current_max + 1, current_max + 1 + count))
    print("Computer played:", ' '.join(map(str, numbers)))
    return numbers

def play_game():
    current_max = 0

    while current_max < 20:

        player_numbers = player_turn(current_max)
        current_max += len(player_numbers)
        if current_max >= 20:
            print("Player Wins!!!")
            break



        computer_numbers = computer_turn(current_max)
        current_max += len(computer_numbers)
        if current_max >= 20:
            print("Computer Wins!!!")
            break

play_game()
```

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

Source code

```
def factorial(num):
    if num == 0 or num == 1:
        return 1
    else:
        return num * factorial(num - 1)

def ncr(n, r):
    if r > n or r < 0:
        return 0
    return factorial(n) // (factorial(r) * factorial(n - r))

def pascaltriangle(rows):
    for n in range(rows):
        for r in range(n + 1):
            print(ncr(n, r), end=' ')
        print()


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

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
```

Source code:

```
n = input("Enter numbers separated by spaces: ")

lst = list(map(int, n.split()))


freq = {}

for num in lst:

    freq[num] = freq.get(num, 0) + 1


for num, count in freq.items():

    print(f"Element {num} has come {count} 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.

Source code:

Matrices.txt

```
1 2

3 4

5 6

7 8
```

```
f = open('matrices.txt', 'r')

A = [list(map(int, f.readline().split())) for i in range(2)]

B = [list(map(int, f.readline().split())) for j in range(2)]

f.close()
```

```
result = [[A[i][j] + B[i][j] for j in range(2)] for i in range(2)]

for row in result:
    print(' '.join(map(str, row)))
```

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

Source code:

```
class Fraction:
    def __init__(self, p, q):
        if q == 0:
            raise ValueError("Denominator cannot be zero")
        self.p, self.q = p, q

    def __add__(self, other):
        n = self.p * other.q + other.p * self.q
        d = self.q * other.q
        return Fraction(n, d).reduce()

    def reduce(self):
        gcd = self._gcd(self.p, self.q)
        return Fraction(self.p // gcd, self.q // gcd)

    @staticmethod
    def _gcd(a, b):
        while b:
            a, b = b, a % b
        return abs(a)

    def __str__(self):
        return f"{self.p}/{self.q}"


f1 = Fraction(1, 2)
f2 = Fraction(3, 4)
result = f1 + f2
print(f"{f1} + {f2} = {result}")
```