**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!!!

Solution For Q1:

```
iimport random

def player_turn(current_num):

    player_choice = input(f"Enter 1 or 2 or 3 numbers from {current_num +1} : ").split()
    player_choice = [int(i) for i in player_choice]

    current_num = player_choice[-1]
    print("Player played : ",player_choice)
    return current_num

def computer_turn(current_num):
    computer_choice = random.randint(1,3)
    computer_number = list(range(current_num+1,current_num+1+computer_choice))
    current_num = computer_number[-1]
    print("Computer played : ",computer_number)
    return current_num

def game():
    current_num = 0

    while current_num < 20:
        current_num = player_turn(current_num)
        if current_num >= 20:
            print("Player WON ")
            break
```

```
        current_num = computer_turn(current_num)
        if current_num >= 20:
            print("computer WON")
            break

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

Solution for Q2:

```
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)
```

**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

Solution For Q3:

```
#list1 = [2,1,2,3,4,5,1,3,6,2,3,4]
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():
    print('Element' ,i ,'has come', j ,'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.

Solution for Q4:

matrices.txt

1 2

3 4

5 6

7 8

```
#matric1 = [[1,2],[3,4]]
#matric2 = [[5,6],[7,8]]

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

    matrix1 = [list(map(int,lines[0].strip().split())),list(map(int,lines[1].strip().split()))]
    matrix2 = [list(map(int,lines[2].strip().split())),list(map(int,lines[3].strip().split()))]

result = [[0,0],[0,0]]
for i in range(2):
    for j in range(2):
        result[i][j] = matrix1[i][j]+matrix2[i][j]

for i in result:
    print(i)
```

**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

Solution for Q5:

```
class addition:
    def __init__(self,p,q):
        self.p = p
```

```
        self.q = q
    def display(self):
        print( self.p ,"/" ,self.q )
    def __add__(self,function):
        fun1 = self.p * function.q + function.p * self.q
        fun2 = (self.q*function.q)
        print(fun1)
        print(fun2)
        return f"{fun1}/{fun2}"

ob1 = addition(1,2)
ob2 = addition(1,3)

result = ob1+ob2

print(result)
```