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!!!

Program code:

```
import random
def computer_turn(current):
    choi = random.choice([1, 2, 3])
    max_reach = min(20, current + 3)
    return list(range(current + 1, current+choi+1))

def player_turn(current):
    while True:
        try:
            player_input = input(f"Your turn (enter 1 to 3 numbers after {current}): ").split()
            player_numbers = list(map(int, player_input))
            if len(player_numbers) not in {1, 2, 3}:
                print("Enter exactly 1, 2, or 3 numbers.")
                continue
            if player_numbers != list(range(current + 1, current + 1 + len(player_numbers))):
                print(f"Numbers must start from {current + 1} and be consecutive.")
                continue
            return player_numbers
        except ValueError:
            print("Invalid input. Please enter numbers only.")
```

```
def play_game():
    current = 0
    while current < 20:
        player_move = player_turn(current)
        current = player_move[-1]
        print(f"You played: {player_move}")
        if current == 20:
            print("Congratulations! You win!")
            break
        computer_move = computer_turn(current)
        current = computer_move[-1]
        print(f"Computer played: {computer_move}")
        if current == 20:
            print("Computer wins! Better luck next time.")
            break
play_game()
```

Output:

```
Your turn (enter 1 to 3 numbers after 0): 1 2 3
You played: [1, 2, 3]
Computer played: [4]
Your turn (enter 1 to 3 numbers after 4): 5 8
Numbers must start from 5 and be consecutive.
Your turn (enter 1 to 3 numbers after 4): 5 6 7
You played: [5, 6, 7]
Computer played: [8, 9, 10]
Your turn (enter 1 to 3 numbers after 10): 11 12 13
You played: [11, 12, 13]
Computer played: [14, 15]
Your turn (enter 1 to 3 numbers after 15): a
Invalid input. Please enter numbers only.
Your turn (enter 1 to 3 numbers after 15): 21
Numbers must start from 16 and be consecutive.
Your turn (enter 1 to 3 numbers after 15): 16
You played: [16]
Computer played: [17, 18, 19]
Your turn (enter 1 to 3 numbers after 19): 20
You played: [20]
```

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 code:

```
import math

def ncr(n, r):
    return math.comb(n, r)

def pascals_triangle(rows):
    for i in range(rows):
        print(" " * (rows - i), end="")

        for j in range(i + 1):
            print(ncr(i, j), end=" ")

        print()

rows = int(input("Enter the number of rows: "))
pascals_triangle(rows)
```

Output:

```
Enter the number of rows: 5
     1
    1 1
   1 2 1
  1 3 3 1
 1 4 6 4 1
```

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

Program code:

```
from collections import Counter
numbers = list(map(int, input("Enter numbers separated by spaces: ").split()))
frequency = Counter(numbers)

for element, count in frequency.items():
    print(f"Element {element} has come {count} times")
```

Output:

```
Enter numbers separated by spaces: 1 2 3 2 2 5 4 3 1
Element 1 has come 2 times
Element 2 has come 3 times
Element 3 has come 2 times
Element 5 has come 1 times
Element 4 has come 1 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.

Program code:

```
print("enter first matrix");
x1=[list(map(int,input().split())) for i in range(2)]
print("enter second matrix")
x2=[list(map(int,input().split())) for j in range(2)]
for i in range(2):
    for j in range(2):
        x1[i][j]+=x2[i][j]
print("Your sum matrix:")
```

```
for row in x1:
    print(*row)
```

Output:

```
enter first matrix
1 2
3 4
enter second matrix
5 6
7 8
Your sum 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 code:

```
from math import gcd


class Fraction:
    def __init__(self, numerator, denominator):
        self.numerator = numerator
        self.denominator = denominator
        self.simplify()

    def simplify(self):
        common_divisor = gcd(self.numerator, self.denominator)
        self.numerator //= common_divisor
        self.denominator //= common_divisor

    def __add__(self, other):
```

```
        # Formula: a/b + c/d = (a * d + b * c) / (b * d)
        new_numerator = self.numerator * other.denominator + other.numerator * self.denominator
        new_denominator = self.denominator * other.denominator
        return Fraction(new_numerator,new_denominator)
    def __str__(self):
        return f"{self.numerator}/{self.denominator}"


f1 = Fraction(1, 2)
f2 = Fraction(3, 4)
result = f1 + f2
print("Result of addition:", result)
```

Output:

```
Result of addition: 5/4
```