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:

```
import random
def computer_move(current_number):
    """Computer picks 1, 2, or 3 numbers in sequence starting from current_number."""
    max_pick = min(3, 20 - current_number)
    computer_choice = random.randint(1, max_pick)
    return [current_number + i for i in range(1, computer_choice + 1)]
def user_move(current_number):
    """Prompt user to enter 1, 2, or 3 numbers in sequence starting from current_number."""
    while True:
        try:
            user_input = input(f"Enter 1, 2, or 3 numbers in sequence starting from {current_number + 1}: ")
            user_numbers = list(map(int, user_input.split()))

            # Validate user input
            if not (1 <= len(user_numbers) <= 3):
                raise ValueError("You must pick between 1 and 3 numbers.")
            if user_numbers[0] != current_number + 1 or any(
                    user_numbers[i] != user_numbers[i - 1] + 1 for i in range(1, len(user_numbers))):
                raise ValueError("Numbers must be sequential.")
            if user_numbers[-1] > 20:
                raise ValueError("Numbers cannot exceed 20.")
```

```
            return user_numbers
        except ValueError as e:
            print(e)

def play_game():
    current_number = 0
    print("Welcome to the Number Game! The goal is to reach 20.")

    while current_number < 20:
        # User's turn
        user_numbers = user_move(current_number)
        current_number = user_numbers[-1]
        print(f"You picked: {user_numbers}")

        if current_number >= 20:
            print("Congratulations! You reached 20 first and won the game!")
            break

        # Computer's turn
        computer_numbers = computer_move(current_number)
        current_number = computer_numbers[-1]
        print(f"Computer picked: {computer_numbers}")

        if current_number >= 20:
            print("Computer reached 20 first. You lost the game.")
            break

if __name__ == "__main__":
    play_game()
```

OUTPUT:

```
============= RESTART: C:\Users\saikr\OneDrive\Desktop\jntu\ass1.py ============
Welcome to the Number Game! The goal is to reach 20.
Enter 1, 2, or 3 numbers in sequence starting from 1: 1 2 3
You picked: [1, 2, 3]
Computer picked: [4]
Enter 1, 2, or 3 numbers in sequence starting from 5: 5
You picked: [5]
Computer picked: [6]
Enter 1, 2, or 3 numbers in sequence starting from 7: 7 8 9
You picked: [7, 8, 9]
Computer picked: [10]
Enter 1, 2, or 3 numbers in sequence starting from 11: 11 12
You picked: [11, 12]
Computer picked: [13, 14, 15]
Enter 1, 2, or 3 numbers in sequence starting from 16: 16
You picked: [16]
Computer picked: [17, 18]
Enter 1, 2, or 3 numbers in sequence starting from 19: 19 20
You picked: [19, 20]
Congratulations! You reached 20 first and won the 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

PROGRAM:

```
def fact(n):
    i = 1
    f = 1
    while (i <= n):
        f = f * i
        i = i + 1
    return f

# nCr function
def ncr(n, r):
    temp1 = fact(n)
    temp2 = fact(n - r)
    temp3 = fact(r)
    return temp1 // (temp2 * temp3)
#pasacal
n = int(input("Enter no. of rows: "))
for i in range(n):
    for j in range(i + 1):
        print(ncr(i, j), end=' ')
    print()
```

OUTPUT:

```
============= RESTART: C:/Users/saikr/OneDrive/Desktop/jntu/ass2.py ============
Enter no. of rows: 3
1
1 1
1 2 1

============= RESTART: C:/Users/saikr/OneDrive/Desktop/jntu/ass2.py ============
Enter no. of rows: 4
1
1 1
1 2 1
1 3 3 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:

```
from collections import Counter
#input
n = int(input("Enter the number of elements in the list: "))
numbers = []
#n inputs
for i in range(n):
    num = int(input(f"Enter number {i + 1}: "))
    numbers.append(num)
# Counting the frequency
frequency_count = Counter(numbers)
# Printing repeated elements
print("\nRepeated elements with their frequency count:")
for num, count in frequency_count.items():
    if count > 1:
        print(f"{num}: {count} times")
```

OUTPUT:

```
============ RESTART: C:/Users/saikr/OneDrive/Desktop/jntu/ass3.py ============
Enter the number of elements in the list: 5
Enter number 1: 1
Enter number 2: 2
Enter number 3: 1
Enter number 4: 3
Enter number 5: 3

Repeated elements with their frequency count:
1: 2 times
3: 2 times
>
```

Question 4:-

a Develop 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(file):
    """Read a 2x2 matrix from the file."""
    matrix = []
    for _ in range(2):  # Read 2 rows
        row = list(map(int, file.readline().strip().split()))
        matrix.append(row)
    return matrix


def add_matrices(matrix_a, matrix_b):
    """Add two 2x2 matrices."""
    return [[matrix_a[i][j] + matrix_b[i][j] for j in range(2)] for i in range(2)]


def main():
```

```
    # Open the file and read the matrices
    with open('matrices.txt', 'r') as file:
        label_a = file.readline().strip()  # Read label for Matrix A
        matrix_a = read_matrix(file)       # Read Matrix A
        label_b = file.readline().strip()  # Read label for Matrix B
        matrix_b = read_matrix(file)       # Read Matrix B

    # Add matrices A and B
    result_matrix = add_matrices(matrix_a, matrix_b)

    # Print the result
    print(f"Result of {label_a} + {label_b}:")
    for row in result_matrix:
        print(row)

if __name__ == "__main__":
    main()
```

MATRICES.TXT:

```
Matrix A
1 2
3 4
Matrix B
5 6
7 8
```

OUTPUT:

```
========== RESTART: C:\Users\saik
Result of Matrix A + Matrix B:
[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:

```
import math
class Fraction:
    def __init__(self, numerator, denominator):
        if denominator == 0:
            raise ValueError("Denominator cannot be zero.")
        self. numerator = numerator
        self. denominator = denominator
        self.simplify()
    def simplify(self):
        gcd = math. gcd(self. numerator, self. denominator)
        self. numerator //= gcd
        self. denominator //= gcd
    def __add__(self, other):
        if not isinstance(other, Fraction):
            raise TypeError("Can only add Fraction objects")

# P1/Q1 + P2/Q2 Formula


        numerator = self. numerator * other. denominator + other. numerator * self. denominator
        denominator = self. denominator * other. denominator
        return Fraction(numerator,denominator)
    def __str__(self):
        return f"{self. numerator}/{self. denominator}"
#main
```

fraction1 = Fraction(1, 2)

fraction2 = Fraction(1, 3)

result = fraction1 + fraction2

print(result)

OUTPUT: