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

```
def check_consecutive(numbers):
    """Check if the list contains consecutive numbers starting from 1."""
    return all(numbers[i] == numbers[i - 1] + 1 for i in range(1, len(numbers)))

def play_number_game():
    print("Number game between user and computer")
    current_number = 0  # Tracks the current number
    max_limit = 20  # Winning number

    while current_number < max_limit:
        # Player's turn
        user_input = input("\nYour turn. Enter 1, 2, or 3 consecutive numbers starting from the next
number: ")
        user_numbers = list(map(int, user_input.split()))

        # Validate player's input
        if len(user_numbers) < 1 or len(user_numbers) > 3 or user_numbers[0] != current_number
+ 1 or not check_consecutive(user_numbers):
            print("Invalid input. You lose!")
            return
        current_number = user_numbers[-1]

        # Check if user wins
        if current_number >= max_limit:
            print("Congratulations, you win!")
            return

        # Computer's turn
```

```
        computer_pick = min(3, max_limit - current_number)
        computer_numbers = list(range(current_number + 1, current_number + computer_pick +
1))
        print("Computer played:", computer_numbers)
        current_number = computer_numbers[-1]

        # Check if computer wins
        if current_number >= max_limit:
            print("Computer wins!")
            return

# Run the game
if __name__ == "__main__":
    play_number_game()
```

Output screenshot

```
Number game between user and computer

Your turn. Enter 1, 2, or 3 consecutive numbers starting from the next number:  1 2 3
Computer played: [4, 5, 6]

Your turn. Enter 1, 2, or 3 consecutive numbers starting from the next number:  7 8 9
Computer played: [10, 11, 12]

Your turn. Enter 1, 2, or 3 consecutive numbers starting from the next number:  13 14
Computer played: [15, 16, 17]

Your turn. Enter 1, 2, or 3 consecutive numbers starting from the next number:  18 19 20
Congratulations, you win!
```

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

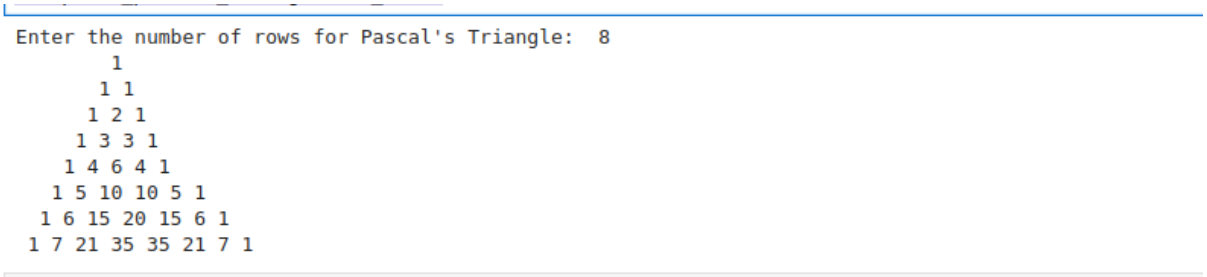
```
def factorial(n):
    """Calculate the factorial of a number."""
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result
```

```
def ncr(n, r):
    """Calculate nCr (combinations)."""
    return factorial(n) // (factorial(r) * factorial(n - r))

def print_pascals_triangle(rows):
    """Print Pascal's Triangle with the given number of rows."""
    for i in range(rows):
        print(" " * (rows - i), end="")
        for j in range(i + 1):
            print(ncr(i, j), end=" ")
        print()

# Get input and print Pascal's Triangle
if __name__ == "__main__":
    num_rows = int(input("Enter the number of rows for Pascal's Triangle: "))
    print_pascals_triangle(num_rows)
```

Output screenshot



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

```
def count_frequencies(numbers):
    """Count the frequency of each element in the list."""
    frequency = {}
    for num in numbers:
        frequency[num] = frequency.get(num, 0) + 1

            # Print each element and its frequency
    for element, count in frequency.items():
        print(f"Element {element} has come {count} times")

# Get input list
if __name__ == "__main__":
    user_input = input("Enter a list of numbers separated by commas: ")
    numbers = list(map(int, user_input.split(',')))
    count_frequencies(numbers)
```

Output screenshot

```
Enter a list of numbers separated by commas:  2,1,3,1,4,5,6,3,8,6,9,6,5,4,6,8,9,1,9,6
Element 2 has come 1 times
Element 1 has come 3 times
Element 3 has come 2 times
Element 4 has come 2 times
Element 5 has come 2 times
Element 6 has come 5 times
Element 8 has come 2 times
Element 9 has come 3 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.

```
def read_matrix_from_file(filename):
    matrices = {}
    current_matrix = None
```

```
    with open(filename, 'r') as file:
        for line in file:
            line = line.strip()
            if line.startswith('A='):
                current_matrix = 'A'
                matrices[current_matrix] = []  # Initialize matrix A
                continue
            elif line.startswith('B='):
                current_matrix = 'B'
                matrices[current_matrix] = []  # Initialize matrix B
                continue

            if current_matrix:
                row = list(map(int, line.split()))
                matrices[current_matrix].append(row)

    return matrices.get('A'), matrices.get('B')

def add_matrices(matrix_a, matrix_b):
    if len(matrix_a) != len(matrix_b) or any(len(row_a) != len(row_b) for row_a, row_b in
zip(matrix_a, matrix_b)):
        raise ValueError("Matrices must be of the same dimensions for addition.")

    result = [[matrix_a[i][j] + matrix_b[i][j] for j in range(len(matrix_a[0]))] for i in
range(len(matrix_a))]
    return result

def print_matrix(matrix, name):
    print(f"Matrix {name}:")
    for row in matrix:
        print(" ".join(map(str, row)))
    print()

if __name__ == "__main__":
    filename = 'Matrices.txt'
    matrix_a, matrix_b = read_matrix_from_file(filename)

    if matrix_a and matrix_b:
        print_matrix(matrix_a, 'A')
        print_matrix(matrix_b, 'B')

        try:
            result_matrix = add_matrices(matrix_a, matrix_b)
            print("Resultant Matrix after addition:")
```

```
            print_matrix(result_matrix, 'Result')
        except ValueError as e:
            print(e)
    else:
        print("Matrices A and/or B were not found in the file.")
```

Output screenshot

```
Matrix A:
3 2
5 4

Matrix B:
1 6
7 4

Resultant Matrix after addition:
Matrix Result:
4 8
12 8
```

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

```
# Function to calculate the Greatest Common Divisor
def gcd(a, b):
    if a == 0:
        return b
    return gcd(b % a, a)

# Function to convert the obtained fraction into its simplest form
def lowest(den3, num3):
    common_factor = gcd(num3, den3)
    den3 = den3 // common_factor  # Use integer division
    num3 = num3 // common_factor
    return num3, den3  # Return the simplified fraction

# Function to add two fractions
def addFraction(num1, den1, num2, den2):
    den3 = gcd(den1, den2)
    den3 = (den1 * den2) // den3  # Use integer division
    num3 = (num1 * (den3 // den1)) + (num2 * (den3 // den2))  # Use integer division
    return lowest(den3, num3)  # Return the simplified fraction
```

```
if __name__ == "__main__":
    # Input first fraction
    num1 = int(input("Enter the numerator of the first fraction: "))
    den1 = int(input("Enter the denominator of the first fraction: "))

    # Input second fraction
    num2 = int(input("Enter the numerator of the second fraction: "))
    den2 = int(input("Enter the denominator of the second fraction: "))

    print(f"{num1}/{den1} + {num2}/{den2} is equal to ", end="")
    result_num, result_den = addFraction(num1, den1, num2, den2)
    print(f"{result_num}/{result_den}")
```

Output screenshot

```
Enter the numerator of the first fraction:  2
Enter the denominator of the first fraction:  4
Enter the numerator of the second fraction:  3
Enter the denominator of the second fraction:  5
2/4 + 3/5 is equal to 11/10
```