Name: VEMULA SHALINI

H.No: 2406DGAL122

****

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

Code 1:

```
import random


def computer_play(current_number):
    # Computer chooses 1 to 3 numbers in sequence
    count = random.randint(1, 3)
    moves = list(range(current_number + 1, current_number + count + 1))
    return moves


def user_input(current_number):
    while True:
        try:
            user_moves = list(map(int, input("Your turn (enter 1 to 3 numbers in sequence): ").split()))
            # Check if moves are in the correct sequence
```

```
            if len(user_moves) >= 1 and len(user_moves) <= 3 and user_moves[0] ==
current_number + 1 and all(
                user_moves[i] == user_moves[i - 1] + 1 for i in range(1, len(user_moves))
            ):
                return user_moves
            else:
                print(f"Please enter 1 to 3 sequential numbers starting from {current_number + 1}.")
        except ValueError:
            print("Invalid input, please enter numbers only.")


def play_game():
    current_number = 0
    while current_number < 20:
        # User's turn
        user_moves = user_input(current_number)
        current_number = user_moves[-1]
        print(f"Player played: {user_moves}")

        if current_number >= 20:
            print("Player Wins!!!")
            break

        # Computer's turn
        computer_moves = computer_play(current_number)
        current_number = computer_moves[-1]
        print(f"Computer played: {computer_moves}")

        if current_number >= 20:
            print("Computer Wins!!!")
```

```
        break

play_game()
```

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!!!

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.

```
# Function to calculate nCr (combinations)
def factorial(x):
    if x == 0 or x == 1:
        return 1
    else:
        result = 1
        for i in range(2, x + 1):
            result *= i
        return result

def ncr(n, r):
    return factorial(n) // (factorial(r) * factorial(n - r))

# Function to print Pascal's Triangle
def print_pascals_triangle(rows):
    for n in range(rows):
        row = []
        for r in range(n + 1):
            row.append(ncr(n, r))
        # Print row centered to form a triangle shape
        print(" " * (rows - n), *row)

# Input number of rows for Pascal's Triangle
rows = int(input())
print_pascals_triangle(rows)
```

sample input: 5

output:

```
     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.

Code 3:

```
arr = eval(input())
mapp = {}
for i in arr:
   mapp[i] = mapp.get(i,0) + 1
for key, val in mapp.items():
   print(f'Elements {key} has come {val} times')
```

Example :

Sample 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

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.

Code 4:

```
def read_matrix():
    matrix = []
    print("Enter the matrix (2x2):")
    for _ in range(2):
        row = list(map(int, input().split()))
        matrix.append(row)
    return matrix
# Function to add two 2x2 matrices
def add_matrices(A, B):
    result = [[0, 0], [0, 0]]
    for i in range(2):
        for j in range(2):
            result[i][j] = A[i][j] + B[i][j]
    return result
# Function to print a matrix
def print_matrix(matrix):
    for row in matrix:
        print(row)
# Read matrices from user input
print("Matrix A:")
matrix_A = read_matrix()
print("Matrix B:")
matrix_B = read_matrix()


# Add matrices A and B
```

```
result_matrix = add_matrices(matrix_A, matrix_B)
# Print the result
print("\nResult of A + B:")
print_matrix(result_matrix)
```

sample input:

2 5

4 10

15 4

10 5

Output:

Matrix A:

Enter the matrix (2x2):

Matrix B:

Enter the matrix (2x2):

Result of A + B:

[17, 9]

[14, 15]

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 5:

```
from math import gcd
class Fraction:
    def __init__(self, numerator, denominator):
        self.numerator = numerator
        self.denominator = denominator
        self.simplify()

    def __add__(self, other):
        # Find the numerator and denominator of the result
        new_numerator = self.numerator * other.denominator + other.numerator * self.denominator
        new_denominator = self.denominator * other.denominator
        return Fraction(new_numerator, new_denominator)

    def simplify(self):
        # Simplify the fraction by dividing by the greatest common divisor
        common_divisor = gcd(self.numerator, self.denominator)
        self.numerator //= common_divisor
        self.denominator //= common_divisor

    def __str__(self):
        # Return a string representation of the fraction
        return f"{self.numerator}/{self.denominator}"
```

Sample input:

```
frac1 = Fraction(1, 2)

frac2 = Fraction(1, 3)

result = frac1 + frac2


print("Result of addition:", result)
```

Output: Result of addition: 5/6