Name of the Candidate: **ABDUL RAHIM**

Name of the Course: **Data Science and Gen Al LLMs**

Email id: **rahim.mrecw@gmail.com** **Contact No:9959324252**

Hall Ticket No: **2406DGAL110** **Assignment:01 Date:21-10-2024**

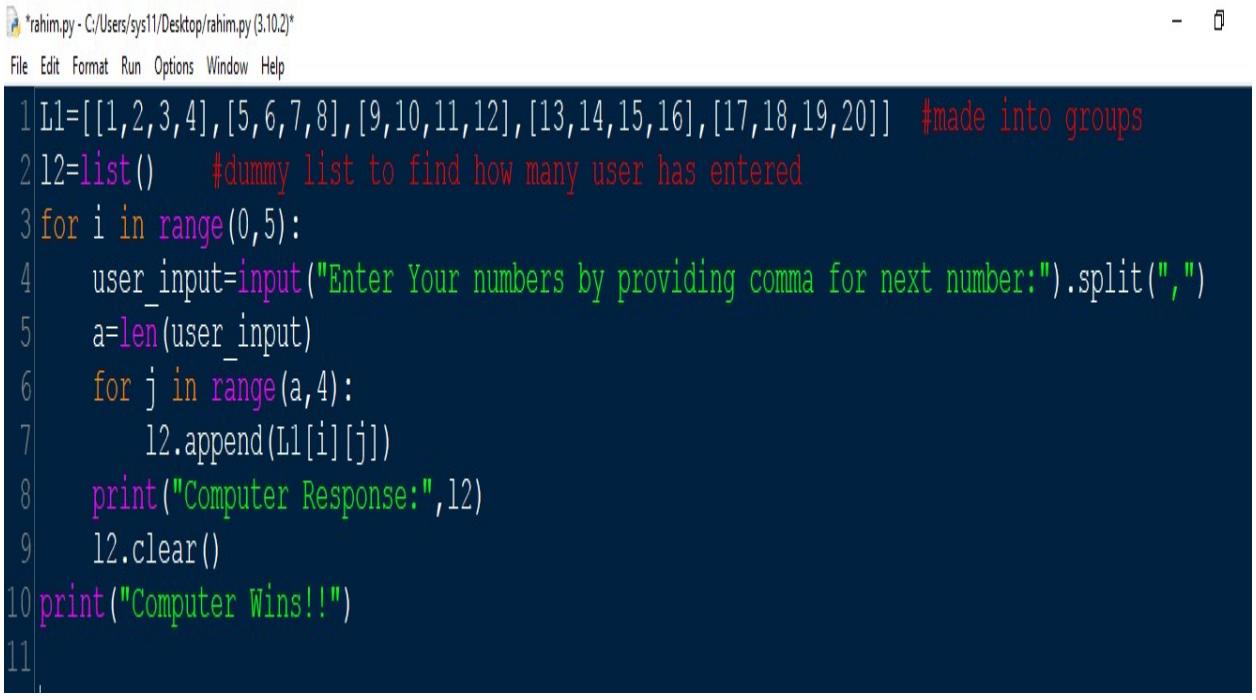
1Q: Number game between user and computer. The user starts by entering either 1 or 2 or 3 digits starting from 1 sequentially. The computer can return either 1 or 2 or 3 next digits in sequence, starting from the max number played by the user. User enters the next 1 or 2 or 3 next digits in sequence, starting from the max number played by the computer. Whoever reaches 20 first wins the game.

ANS:

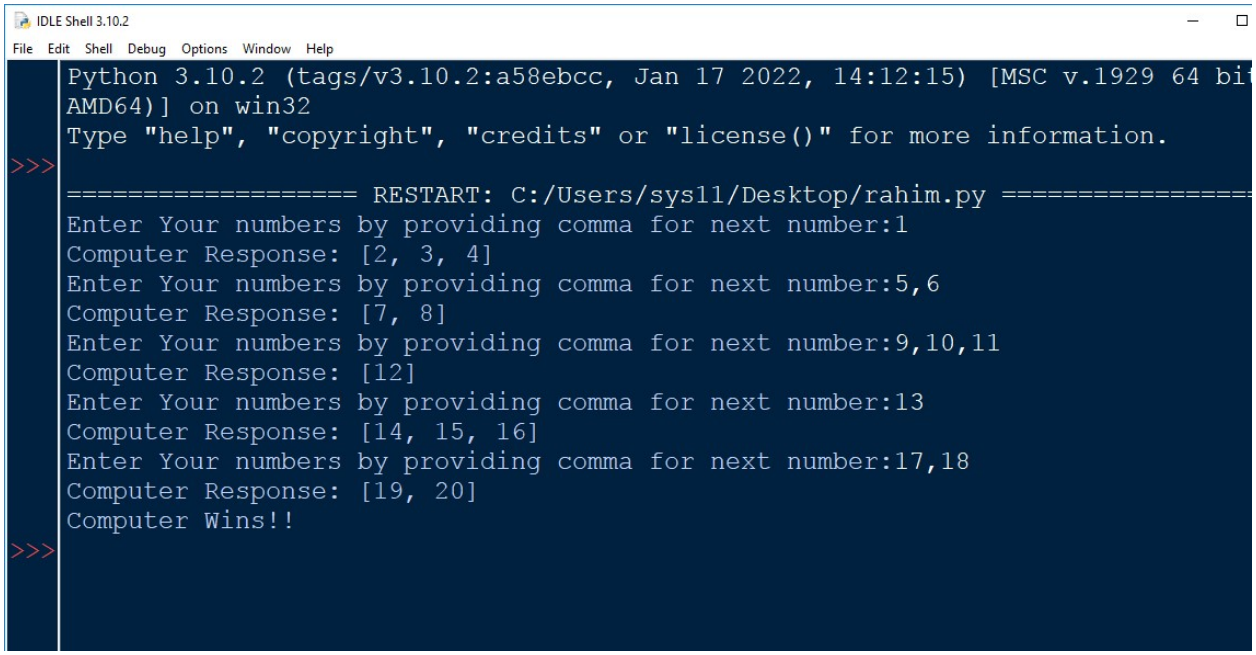
#i have taken a sequence as a list where 20 numbers are divided into groups of 4.

#i then compared with how many user is providing, depending on that, program will select one #or two or three such that every time the computer will win.

PROGRAM:



OUTPUT:



2Q: 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

ANS:

Pascal's Triangle is the triangular arrangement of numbers that gives the coefficients in the expansion of any binomial expression. The numbers are so arranged that they reflect as a triangle.

Technique: The easiest way to construct the triangle is to start at row zero and write only the number one. From there, to obtain the numbers in the following rows, add the number directly above and to the left of the number with the number above and to the right of it. If there are no numbers on the left or right side, replace a zero for that missing number and proceed with the addition.

PROGRAM:

rahim1.py - C:/Users/sys11/Desktop/rahim1.py (3.10.2)

File Edit Format Run Options Window Help

```
def ncr(n, r):
    if r > n or r < 0:
        return 0
    numerator = 1
    denominator = 1
    result=numerator // denominator
    for i in range(1, r + 1):
        numerator *= (n - i + 1)
        denominator *= i
        result=numerator // denominator
    return result
def print_pascals_triangle(rows):
    for i in range(rows):
        print(' ' * (rows - i), end='')
        for j in range(i + 1):
            value = ncr(i, j)
            print(value, end=' ')

        print()


num_rows = int(input("Enter the number of rows for Pascal's Triangle: "))
print_pascals_triangle(num_rows)
```

OUTPUT:

IDLE Shell 3.10.2

File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/sys11/Desktop/rahim1.py ==========
Enter the number of rows for Pascal's Triangle: 6
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
>>>
```

3Q: Read a list of n numbers during runtime. Write a Python program to print the repeated elements with frequency count in a list.

ANS:

I will be taking a dictionary where items will be the provided number and value shows the frequency.

PROGRAM:

rahim1.py - C:/Users/sys11/Desktop/rahim1.py (3.10.2)

File Edit Format Run Options Window Help

```
def count_f(numbers):
    frequency_dict = {}
    for number in numbers:
        if number in frequency_dict:
            frequency_dict[number] += 1
        else:
            frequency_dict[number] = 1
    return frequency_dict
def elements(frequency_dict):
    print("Repeated elements with their frequency count:")
    for number, count in frequency_dict.items():
        if count >=1:
            print(f"Element: {number}, has come: {count} times")


input_numbers = input("Enter a list of numbers separated by spaces: ")
numbers = list(map(int, input_numbers.split()))
frequency_dict = count_f(numbers)
elements(frequency_dict)

```

OUTPUT:

IDLE Shell 3.10.2

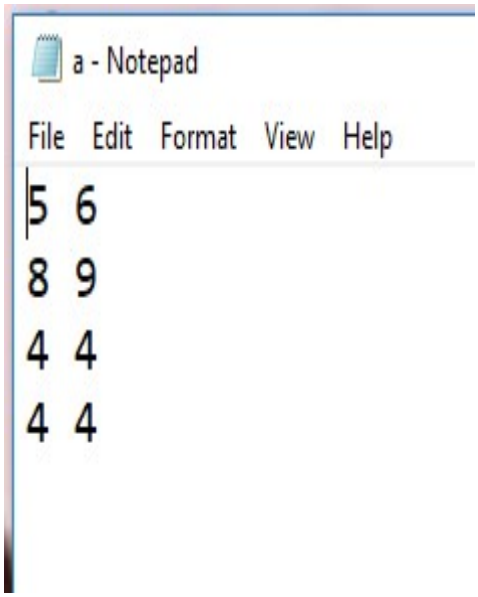
File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/sys11/Desktop/rahim1.py ==========
Enter a list of numbers separated by spaces: 3 4 5 5 5 7 8 10 12 12 7
Repeated elements with their frequency count:
Element: 3, has come: 1 times
Element: 4, has come: 1 times
Element: 5, has come: 3 times
Element: 7, has come: 2 times
Element: 8, has come: 1 times
Element: 10, has come: 1 times
Element: 12, has come: 2 times
>>>
```

4Q: Develop a python code to read matrix 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.

ANS:

I have taken a note pad and stored the matrix values as follows:



Here I have taken a 2X2 matrix where first two rows are for Matrix A and the next two rows are matrix B. In the program, A range is (0,2) and B range is (2,4) and I will read line by line and append in a list. After that I will add the matrix and store in another text.

PROGRAM:

rahim1.py - C:\Users\sys11\Desktop\rahim1.py (3.10.2)

File Edit Format Run Options Window Help

```
def read_matrices_from_file(filename):
    with open("a.txt", 'r') as file:
        lines = file.readlines()
    A = []
    B = []
    for i in range(2):
        A.append(list(map(int, lines[i].strip().split())))
    for i in range(2, 4):
        B.append(list(map(int, lines[i].strip().split())))
    return A, B
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

def print_matrix(matrix):
    for row in matrix:
        print(' '.join(map(str, row)))
filename = 'matrices.txt'
A, B = read_matrices_from_file(filename)
print("Matrix A:")
print_matrix(A)
print("\nMatrix B:")
print_matrix(B)
result = add_matrices(A, B)
print("\nResult of A + B:")
print_matrix(result)
```

OUTPUT:

IDLE Shell 3.10.2

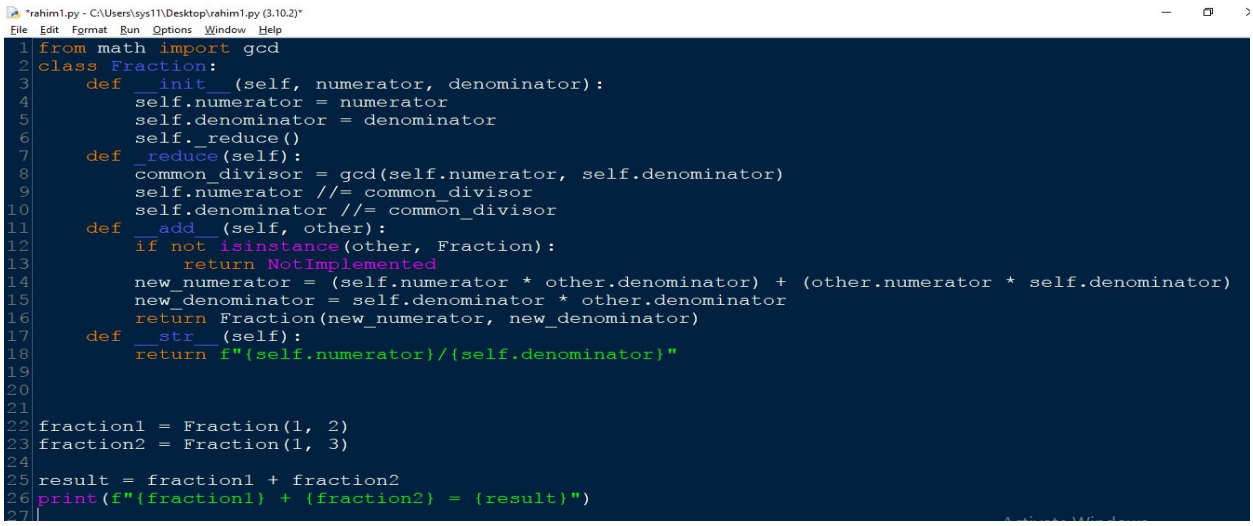
File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\sys11\Desktop\rahim1.py ==========
Matrix A:
5 6
8 9

Matrix B:
4 4
4 4

Result of A + B:
9 10
12 13
>>>
```

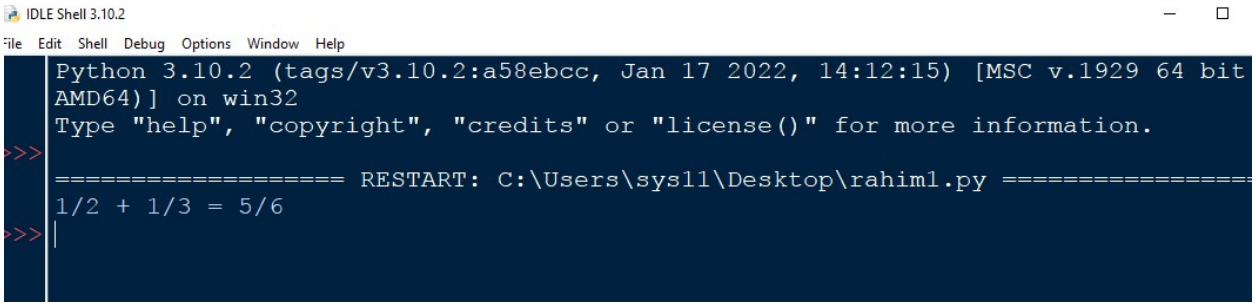
5Q: Write a program that overloads the + operator so that it can add two objects of the class Fraction. Fraction can be considered of the for P/Q where P is the numerator and Q is the denominator

ANS: In this a Fraction class is created, which will have numerator and denominator variables. After that GCD is used in order to reduce the fraction to a simplest form. Addition is performed in order to calculate the new numerator and denominator and returns a new Fraction object. The result will be a user friendly value. In this example I have take 1/ 2 and 1 / 3

PROGRAM:



OUTPUT:



One more example (2/9 and 4/15)