

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!!!

CODE :

```
import random
```

```
def player_turn(current_num):
```

```
    player_choice = input(f'Enter 1 or 2 or 3 numbers from {current_num +1} : ').split()  
    player_choice = [int(i) for i in player_choice]
```

```
    current_num = player_choice[-1]  
    print("Player played : ",player_choice)  
    return current_num
```

```
def computer_turn(current_num):
```

```
    computer_choice = random.randint(1,3)  
    computer_number = list(range(current_num+1,current_num+1+computer_choice))  
    current_num = computer_number[-1]  
    print("Computer played : ",computer_number)  
    return current_num
```

def game():

current_num = 0

while current_num < 20:

current_num = player_turn(current_num)

if current_num >= 20:

print("Player WON ")

break

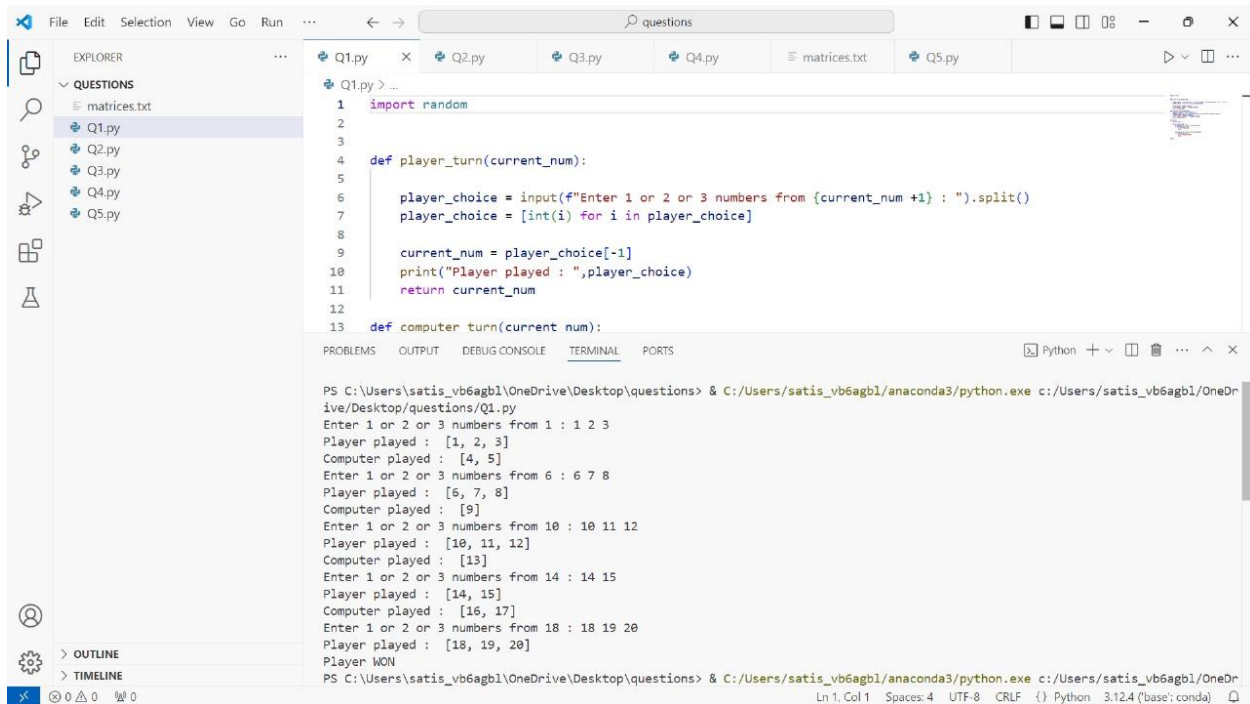
current_num = computer_turn(current_num)

if current_num >= 20:

print("computer WON")

break

game()



The screenshot shows a Python IDE with a file explorer on the left containing a folder named 'QUESTIONS' with files Q1.py through Q5.py and matrices.txt. The main editor displays the code for Q1.py, which defines two functions: `player_turn` and `computer_turn`. The `player_turn` function prompts the user to enter 1, 2, or 3 numbers and returns the list of numbers. The `computer_turn` function is currently empty. The terminal at the bottom shows the execution of `Q1.py`, demonstrating the game's progress: the player enters numbers, the computer plays, and the game continues until the player wins.

```
1 import random
2
3
4 def player_turn(current_num):
5
6     player_choice = input(f"Enter 1 or 2 or 3 numbers from {current_num+1} : ").split()
7     player_choice = [int(i) for i in player_choice]
8
9     current_num = player_choice[-1]
10    print("Player played : ",player_choice)
11    return current_num
12
13 def computer_turn(current num):
```

PS C:\Users\satis_vb6agbl\OneDrive\Desktop\questions> & C:/Users/satis_vb6agbl/anaconda3/python.exe c:/Users/satis_vb6agbl/OneDrive/Desktop/questions/Q1.py
Enter 1 or 2 or 3 numbers from 1 : 1 2 3
Player played : [1, 2, 3]
Computer played : [4, 5]
Enter 1 or 2 or 3 numbers from 6 : 6 7 8
Player played : [6, 7, 8]
Computer played : [9]
Enter 1 or 2 or 3 numbers from 10 : 10 11 12
Player played : [10, 11, 12]
Computer played : [13]
Enter 1 or 2 or 3 numbers from 14 : 14 15
Player played : [14, 15]
Computer played : [16, 17]
Enter 1 or 2 or 3 numbers from 18 : 18 19 20
Player played : [18, 19, 20]
Player WON
PS C:\Users\satis_vb6agbl\OneDrive\Desktop\questions> & C:/Users/satis_vb6agbl/anaconda3/python.exe c:/Users/satis_vb6agbl/OneDr

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

CODE :

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

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

def triangle(r):
    for i in range(r):
        print(' ' * (r - i), end="")
        for j in range(i+1):
            print(ncr(i,j), end=' ')
        print()

n= int(input("Enter n rows : "))
triangle(n)
```

The image shows a Python IDE window with the following components:

- EXPLORER:** A sidebar on the left showing a folder named "QUESTIONS" containing files: matrices.txt, Q1.py, Q2.py (selected), Q3.py, Q4.py, and Q5.py.
- Code Editor:** The main area displays the code for Q2.py:

```
1 def fact(n):
2     i=1
3     f=1
4     while (i<=n):
5         f=f*i
6         i+=1
7     return f
8
9 def ncr(n,r):
10    return fact(n)//(fact(n-r)*fact(r))
11
12 def triangle(r):
13    for i in range(r):
14        print(' ' * (r - i), end='')
15        for j in range(i+1):
16            print(ncr(i,j), end=' ')
17        print()
18
19 n= int(input("Enter n rows : "))
20 triangle(n)
```
- TERMINAL:** The bottom panel shows the execution of the script:

```
PS C:\Users\satis_vb6agbl\OneDrive\Desktop\questions> & C:/Users/satis_vb6agbl/anaconda3/python.exe c:/Users/satis_vb6agbl/OneDrive/Desktop/questions/Q2.py
Enter n rows : 6
 1 1
 1 2 1
 1 3 3 1
 1 4 6 4 1
 1 5 10 10 5 1
```
- STATUS BAR:** The bottom right corner shows "Ln 1, Col 1", "Spaces: 4", "UTF-8", "CRLF", "Python", and "3.12.4 (base: conda)".

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

CODE:

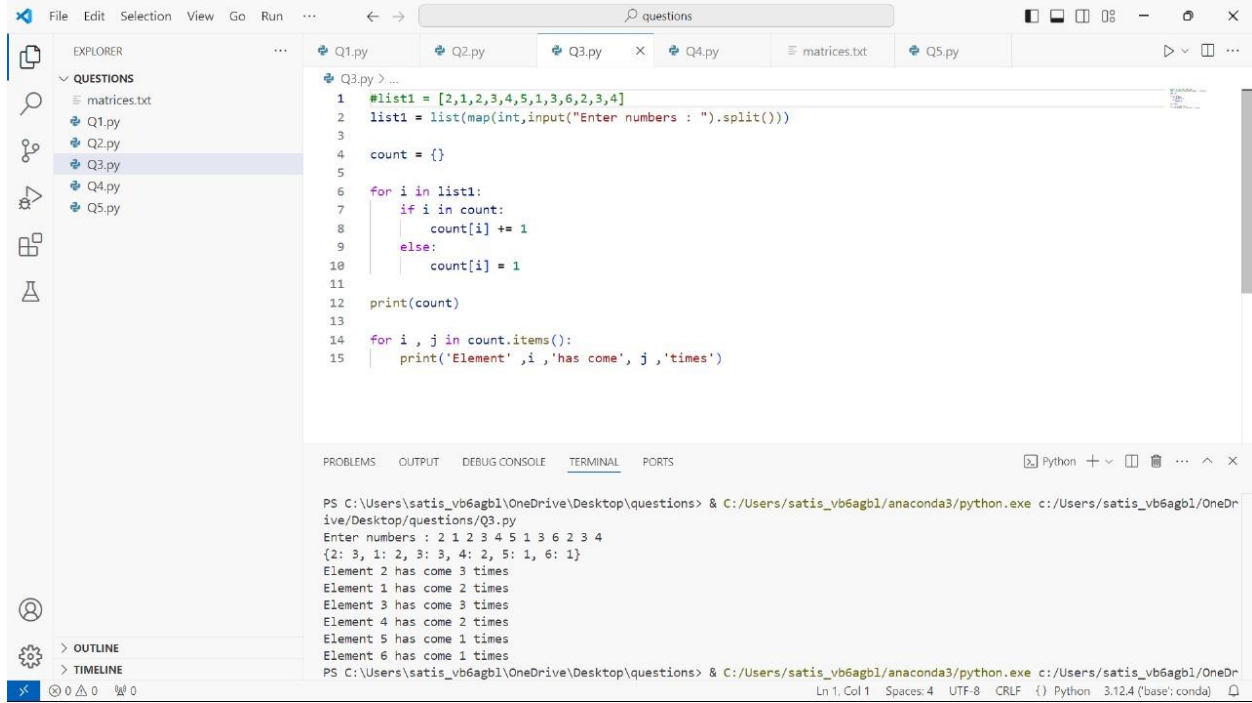
```
#list1 = [2,1,2,3,4,5,1,3,6,2,3,4]
list1 = list(map(int,input("Enter numbers : ").split()))

count = {}

for i in list1:
    if i in count:
        count[i] += 1
    else:
        count[i] = 1

print(count)
```

```
for i, j in count.items():  
    print('Element', i, 'has come', j, 'times')
```



Question 4:-

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.

CODE:

```
#matrix1 = [[1,2],[3,4]]
```

```
#matrix2 = [[5,6],[7,8]]
```

```
with open('matrices.txt','r') as file:
```

```
    lines = file.readlines()
```

```
    matrix1 = [list(map(int,lines[0].strip().split())),list(map(int,lines[1].strip().split()))]
```

```
    matrix2 = [list(map(int,lines[2].strip().split())),list(map(int,lines[3].strip().split()))]
```

```
result = [[0,0],[0,0]]
```

```
for i in range(2):
```

```
    for j in range(2):
```

```
        result[i][j] = matrix1[i][j]+matrix2[i][j]
```

```
for i in result:
```

```
    print(i)
```


The image shows a Python IDE interface with the following components:

- EXPLORER:** A sidebar on the left showing a project named "QUESTIONS" containing files: `matrices.txt`, `Q1.py`, `Q2.py`, `Q3.py`, `Q4.py` (selected), and `Q5.py`.
- Code Editor:** The main window displays the code for `Q4.py`. The code defines two matrices, reads lines from `matrices.txt`, and performs a matrix addition. The code is as follows:

```
1 #matrix1 = [[1,2],[3,4]]
2 #matrix2 = [[5,6],[7,8]]
3
4 with open('matrices.txt','r') as file:
5     lines = file.readlines()
6
7     matrix1 = [list(map(int,lines[0].strip().split())),list(map(int,lines[1].strip().split()))]
8     matrix2 = [list(map(int,lines[2].strip().split())),list(map(int,lines[3].strip().split()))]
9
10
11 result = [[0,0],[0,0]]
12 for i in range(2):
13     for j in range(2):
14         result[i][j] = matrix1[i][j]+matrix2[i][j]
15
16 for i in result:
17     print(i)
```
- TERMINAL:** The terminal at the bottom shows the execution path and the output of the script:

```
ive/Desktop/questions/Q4.py
[6, 8]
[10, 12]
PS C:\Users\satis_vb6agbl\OneDrive\Desktop\questions> & C:/Users/satis_vb6agbl/anaconda3/python.exe c:/Users/satis_vb6agbl/OneDr
```
- STATUS BAR:** The bottom status bar indicates the current cursor position: `Ln 1, Col 1`, `Spaces: 4`, `UTF-8`, `CRLF`, `{}`, `Python`, `3.12.4 (base: conda)`.

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 form P/Q where P is the numerator and Q is the denominator

CODE:

```
class addition:
```

```
    def __init__(self,p,q):
```

```
        self.p = p
```

```
        self.q = q
```

```
    def display(self):
```

```
        print( self.p ,"/" ,self.q )
```

```
    def __add__(self,function):
```

```
        fun1 = self.p * function.q + function.p * self.q
```

```
        fun2 = (self.q*function.q)
```

```
        print(fun1)
```

```
        print(fun2)
```

```
        return f"{fun1}/{fun2}"
```

```
ob1 = addition(1,2)
```

```
ob2 = addition(1,3)
```

```
result = ob1+ob2
```

```
print(result)
```

The image shows a Python IDE interface with the following components:

- EXPLORER:** A sidebar on the left showing a folder named "QUESTIONS" containing files "matrices.txt", "Q1.py", "Q2.py", "Q3.py", "Q4.py", and "Q5.py".
- EDITOR:** The main window displays the code for "Q5.py":

```
1 class addition:
2     def __init__(self,p,q):
3         self.p = p
4         self.q = q
5     def display(self):
6         print( self.p ,"/" ,self.q )
7     def __add__(self,function):
8         fun1 = self.p * function.q + function.p * self.q
9         fun2 = (self.q*function.q)
10        print(fun1)
11        print(fun2)
12        return f"{fun1}/{fun2}"
13
14 ob1 = addition(1,2)
15 ob2 = addition(1,3)
16
17 result = ob1+ob2
18
19 print(result)
```
- TERMINAL:** A terminal window at the bottom shows the command prompt with the current directory: "PS C:\Users\satis_vb6agbl\OneDrive\Desktop\questions>".
- STATUS BAR:** The bottom right corner shows "Ln 19, Col 14", "Spaces: 4", "UTF-8", "CRLF", "Python", and "3.12.4 (base: conda)".