Udaya Kadali (kkadali@gmail.com)

Question 1: Game between computer and player

Code

```
import random

def checkConsecutive(l):
   return sorted(l) == list(range(min(l), max(l)+1))

def enterNumber(iNum):
   validation = False

   while validation == False:
      try:
         sPlayer = input("Player enter number(s) with a space: ")
         sPlayer = sPlayer.strip();

         aNum = sPlayer.split(" ");

         validation = validate(aNum, iNum)

         return max(aNum);
      except ValueError as e:
         print(e)

def validate(aNum, iNum):
   if (len(aNum) > 3):
      raise ValueError ("1 or 2 or 3 numbers are only allowed.")

   for i in range(0, len(aNum)):
      aNum[i] = int(aNum[i])

   if (min(aNum) <= iNum):
      raise ValueError ("Entered number is less than current number! ", iNum)

   if (min(aNum) != iNum+1):
      raise ValueError ("Entered number should start from next sequence! current number is ", iNum)

   isSeq = checkConsecutive(aNum)
   if (isSeq == False):
      raise ValueError ("Enter sequential numbers only!")

   return True;

def computerReturns(i):
   aComp = []

   if (20 - i > 3):
      number = random.randint(1, 3)
```

```
    else:
        number = 20-i

    for j in range(0, number):
        i += 1
        aComp.append(i)
    print ('Computer played: ', aComp)
    return i;

def checkMaxNumber(i):
    if i >= 20:
        return True;

def main():
    iNum = 0

    while iNum < 20:
        iNum = enterNumber(iNum)
        if (checkMaxNumber(iNum) == True):
            print ("Player Wins!!!")
        else:
            iNum = computerReturns(iNum)
            if (checkMaxNumber(iNum) == True):
                print ("Computer Wins!!!")

        if (iNum >= 20):
            exit

main()
```

Results

```
main()
```

```
Player enter number(s) with a space: 1
Computer played:  [2, 3, 4]
Player enter number(s) with a space: 3
('Entered number is less than current number! ', 4)
Player enter number(s) with a space: 5
Computer played:  [6, 7, 8]
Player enter number(s) with a space: 10
('Entered number should start from next sequence! current number is ', 8)
Player enter number(s) with a space: 9
Computer played:  [10]
Player enter number(s) with a space: 11 12 13
Computer played:  [14, 15]
Player enter number(s) with a space: 16 17
Computer played:  [18, 19, 20]
Computer Wins!!!
```

```
]:
```

Udaya Kadali (kkadali@gmail.com)

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 nCr(n, r):
    if r > n:
        return 0
    if r == 0 or r == n:
        return 1
    return nCr(n-1, r-1) + nCr(n-1, r)

rows = int(input("Enter number of rows for the triangle: "))
triangle = [[]]
for i in range(0, rows):
    row = []
    j = 0
    while j <= i:
        row.append(nCr(i, j))
        j+=1
    triangle.append(row)

for row in triangle:
    print (' '.join(map(str, row)).center(rows*3))
```

Results:

In [28]:

```
def nCr(n, r):
    if r > n:
        return 0
    if r == 0 or r == n:
        return 1
    return nCr(n-1, r-1) + nCr(n-1, r)

rows = int(input("Enter number of rows for the triangle: "))
triangle = [[]]
for i in range(0, rows):
    row = []
    j = 0
    while j <= i:
        row.append(nCr(i, j))
        j+=1
    triangle.append(row)

for row in triangle:
    print (' '.join(map(str, row)).center(rows*3))
```

```
Enter number of rows for the triangle: 7

          1
        1 1
       1 2 1
      1 3 3 1
    1 4 6 4 1
  1 5 10 10 5 1
1 6 15 20 15 6 1
```

Udaya Kadali (kkadali@gmail.com)

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

```
inputs = []
while True:
   s = input("Enter number: ").strip()
   if s == "":
      break
   inputs.append(int(s))

print("The numbers entered are: ", inputs)

results = {key: inputs.count(key) for key in inputs}

for key, value in results.items():
   print(f"Element {key} has come {value} times")
```

Output

In [2]:

```
inputs = []
while True:
    s = input("Enter number: ").strip()
    if s == "":
        break
    inputs.append(int(s))

print("The numbers entered are: ", inputs)

results = {key: inputs.count(key) for key in inputs}

for key, value in results.items():
    print(f"Element {key} has come {value} times")
```

```
Enter number: 1
Enter number: 22
Enter number: 3
Enter number: 3
Enter number: 3
Enter number: 2
Enter number: 2
Enter number: 4
Enter number: 5
Enter number:
The numbers entered are:  [1, 22, 3, 3, 3, 2, 2, 4, 5]
Element 1 has come 1 times
Element 22 has come 1 times
Element 3 has come 3 times
Element 2 has come 2 times
Element 4 has come 1 times
Element 5 has come 1 times
```

Udaya Kadali (kkadali@gmail.com)

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:

```
import numpy as np

def readfile(filename):
   with open(filename, 'r') as f:
      matrix = []
      for line in f:
         row = [float(x) for x in line.split()]
         matrix.append(row)
   return np.array(matrix)

try:
   m1 = readfile("Assignment4input1.txt")
   m2 = readfile("Assignment4input2.txt")

   print ("Matrix A: ")
   print (m1)
   print ()
   print ("Matrix B: ")
   print (m2)
   print ()
   print ("Matrix A + Matrix B: ")
   print (m1 + m2)

except Exception as ex:
   print (ex)
```

Results:

```
Matrix A:
[[2. 3.]
 [5. 6.]]

Matrix B:
[[12. 13.]
 [15. 16.]]

Matrix A + Matrix B:
[[14. 16.]
 [20. 22.]]
```

Udaya Kadali (kkadali@gmail.com)

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

```
class Fraction:
   def __init__(self, numerator, denominator):
      self.P = numerator
      self.Q = denominator

   def __add__(self, obj):
      new_P = self.P * obj.Q + obj.P * self.Q
      new_Q = self.Q * obj.Q
      return Fraction(new_P, new_Q)

   def __str__(self):
      return f"{self.P}/{self.Q}"


f1 = Fraction(1,3)
f2 = Fraction(1,4)

f3 = f1 + f2
print (f3)
```

Results:

In [9]:

```
class Fraction:
    def __init__(self, numerator, denominator):
        self.P = numerator
        self.Q = denominator

    def __add__(self, obj):
        new_P = self.P * obj.Q + obj.P * self.Q
        new_Q = self.Q * obj.Q
        return Fraction(new_P, new_Q)

    def __str__(self):
        return f"{self.P}/{self.Q}"


f1 = Fraction(1,3)
f2 = Fraction(1,4)

f3 = f1 + f2
print (f3)
```

```
7/12
```