**ASSIGNMENT 2**

**Algorith: -**

1. Take the quarter of dohas as input.
2. Create a list of scansion of each word, word\_scansion = [scansion of each word as a list]
3. Create a temporary list quarter\_kala = []
4. Count := 0

To check for kalas, first we check if the scansion of first word becomes chuakal or trikal

Then if chaukal is found then we add another chaukal to the list and then add remaining scansion.

If first word is neither trikal nor chaukal then we add the sum of word to a variable named count and check check each scansion of the next word and check if trikal or chaukal is forming

1. For word in word\_scansion:

Checks if the sum of first word is chaukal or not, if chaukal is found then we append C, C in the list

1. If (sum(word) == 4 or count+sum(word) == 4):
2. Quarter\_kala.append(“C”)
3. Quarter\_kala.append(“C”)
4. Break

Check if the word is is foeming a trikal, if yes then append T, T, D

1. Else if (sum(word) == 3)
2. Kala.append(“T”)
3. Kala.append(“T”)
4. Kala.append(“D”)
5. Break

If the word is neither forming Trikala or chaukal then we check each scansion of the word and add it to the variable count and when count becomes 4, then we append C,C or if count is three then we append T,T,D.

1. Else:
2. For letter in word:
3. Count += 1
4. If (count == 4)
5. Quarter\_kala.append(“C”)
6. Quarter\_kala.append(“C”)
7. Break
8. Else if (count == 3):
9. Kala.append(“T”)
10. Kala.append(“T”)
11. Kala.append(“D”)
12. Break

To append rest of the scansion as it is.

1. Iterator := 0
2. Count := 0
3. For word in word:
4. For letter in word:
5. if (total\_count < 8):
6. total\_count += letter
7. iterator += 1
8. else:
9. quarter\_kala.append(letter)
10. iterator += 1
11. total\_count += letter

**Code: -**

#This is for counting kala in a given Doha

class Doha(MatraCount):

    def \_\_init\_\_(self, doha\_line):

        super().\_\_init\_\_(doha\_line)

        self.kala = []

        self.scansion\_quarters = []

        self.word\_scansion\_quarter = []

    #this funciton coiunts the kala of the given lines of doha respectively,

    # first we check if the word is forming a chaukal or trikal

    # if the first word is forming a chaukal then we append "C", "C" tp quarter\_kala list

    # but if the first word has trikal then we append "T", "T", "D"

    # but if no chaukal or trikal is found then we add each matra in a varible count to check

    # for trikala or chaukala, and if trikala or cahukal is found and we append in quarter kala accordingly.

    def countingKala(self, line):

        quarter\_kala = []

        kala\_counter = 0

        found = False

        # This loop iterates through scansion of each word and then check for

        # chaukal and trikal as mentioned above

        for matra\_in\_word in line:

            # to check if fist word is chaukal or the sum of first and second word is chaukal or not

            if ((sum(matra\_in\_word) == 4 or (kala\_counter+sum(matra\_in\_word)) == 4) and found == False):

                quarter\_kala.append("C")

                quarter\_kala.append("C")

                found = True

                break

            # check if the first word is trikal

            elif (sum(matra\_in\_word) == 3 and found == False):

                quarter\_kala.append("T")

                quarter\_kala.append("T")

                quarter\_kala.append("D")

                break

            #  if not trikal found in first word then we go for adding

            elif (found == False):

                for matra in matra\_in\_word:

                    kala\_counter += matra

                    if (kala\_counter == 4):

                        quarter\_kala.append("C")

                        quarter\_kala.append("C")

                        break

                    elif (kala\_counter == 3):

                        quarter\_kala.append("T")

                        quarter\_kala.append("T")

                        quarter\_kala.append("D")

                        break

# the below code appends the remaining matra count as it is

        iterator = 0

        total\_count = 0

        for word in line:

            for  letter in word:

                if (total\_count < 8):

                    total\_count += letter

                    iterator += 1

                else:

                    quarter\_kala.append(letter)

                    iterator += 1

                    total\_count += letter

        return quarter\_kala

# this function break the scansion into quarters.

    def breakScansion(self):

        count = 0

        temp = []

# iterating from backward then when we encounter sum is 11 then we append the matra in temporary list

# and then append it(temporary list) to the list named word\_scansion\_quarter.

        for i in range(len(self.word\_scansion)-1, -1, -1):

            if (count < 11):

                count += sum(self.word\_scansion[i])

                temp.insert(0, self.word\_scansion[i])

                if (count == 11):

                    self.word\_scansion\_quarter.append(temp[:])

                    temp.clear()

            elif (count>=11):

                temp.insert(0, self.word\_scansion[i])

                count += sum(self.word\_scansion[i])

        self.word\_scansion\_quarter.insert(0, temp[:])

        return self.word\_scansion\_quarter

# this function returns the kala generated

    def getKala(self):

        for quarter in self.breakScansion():

            x = self.countingKala(quarter)

            self.kala.append(x[:])

        return self.kala

# Driver Code

lines = []

for i in range(0,2):

inp = input().split(" ")

lines.append(inp)

for i in range (0, 2):

line = Doha(lines[i])

print(line.getKala(), end=" -> ")

print(line.getSumOfMatra())

**A computer screen shot of a computer code

Description automatically generated**