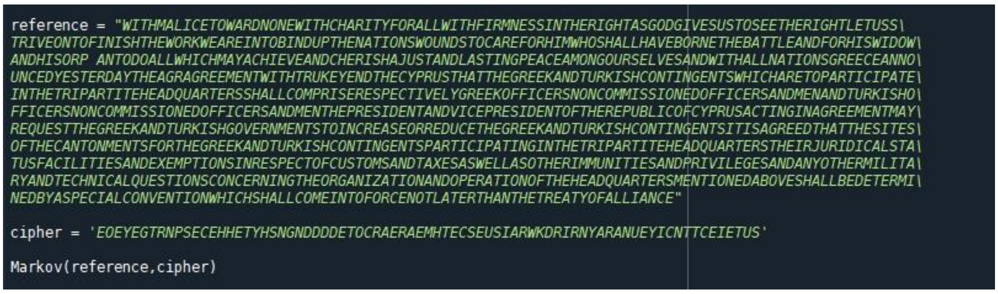
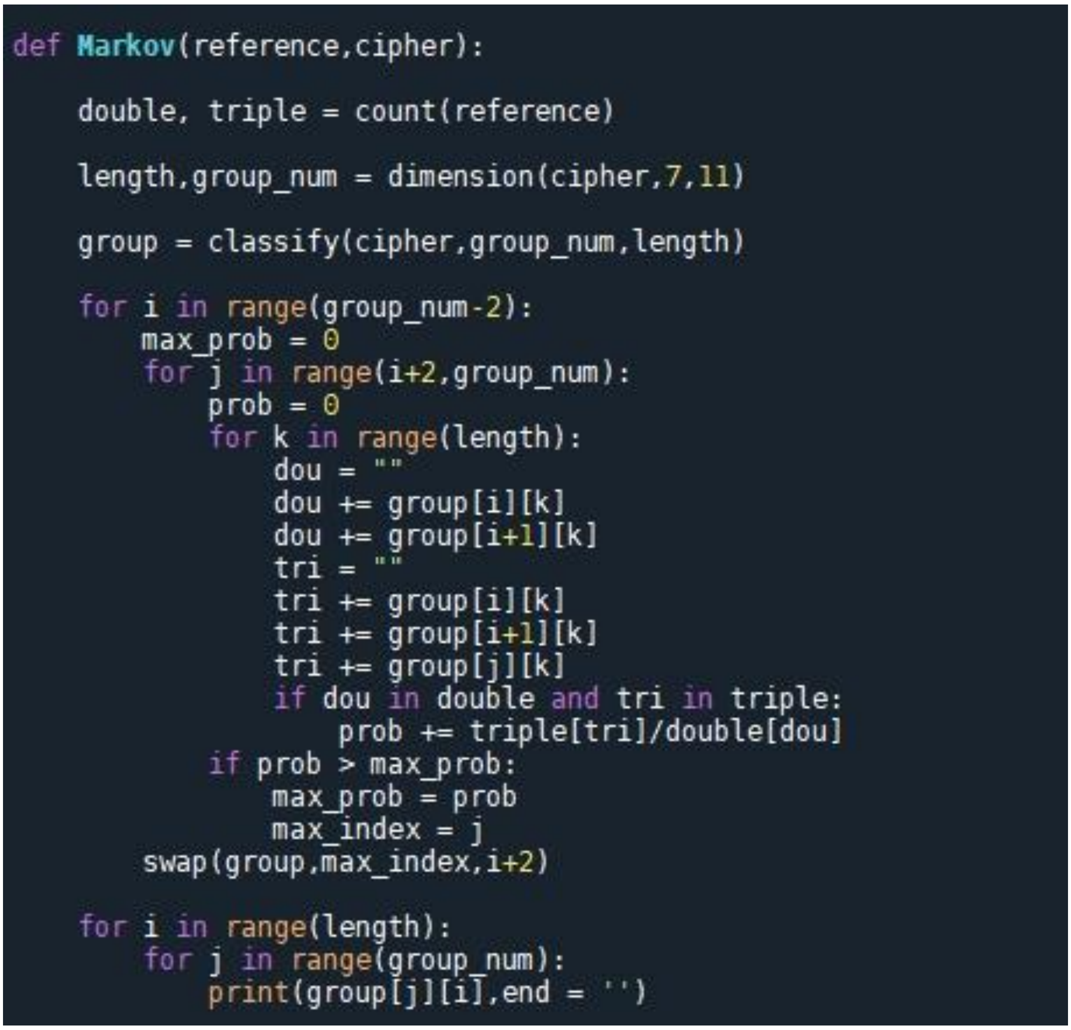
Explanation of my Code

# main: 給 reference 和 cipher



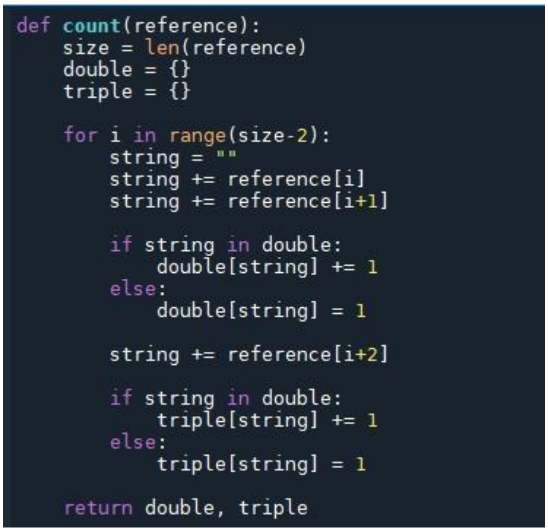
```
reference = "WITHMALICETOWARDNONEWITHCHARITYFORALLWITHFIRMNESSINTHERIGHTASGODGIVESUSTOSEETHERIGHTLETUSS\
TRIVEONTOFINISHTHEWORKWEAREINTOBINDUPTHENATIONSWOUNDSTOCAREFORHIMWHOSHALLHAVEBORNETHEBATTLEANDFORHISWIDOW\
ANDHISORP ANTODOALLWHICHMAYACHIEVEANDCHERISHAJUSTANDLASTINGPEACEAMONGOURSELVESANDWITHALLNATIONSGREECEANNO\
UNCEDYESTERDAYTHEAGRAGREEMENTWITHTRUKEYENDTHECYPRUSTHATTHEGREEKANDTURKISHCONTINGENTSWHICHARETOPARTICIPATE\
INTHETRIPARTITEHEADQUARTERSSHALLCOMPRISERESPECTIVELYGREEKOFFICERSNONCOMMISSIONEDOFFICERSANDMENANDTURKISHO\
FFICERSNONCOMMISSIONEDOFFICERSANDMENTHEPRESIDENTANDVICEPRESIDENTOFTHEREPUBLICOFCYPRUSACTINGINAGREEMENTMAY\
REQUESTTHEGREEKANDTURKISHGOVERNMENTSTOINCREASEORREDUCETHEGREEKANDTURKISHCONTINGENTSITISAGREEDTHATTHESITES\
OFTHECANTONMENTSFORTHEGREEKANDTURKISHCONTINGENTSPARTICIPATINGINTHETRIPARTITEHEADQUARTERSTHEIRJURIDICALSTA\
TUSFACILITIESANDEXEMPTIONSINRESPECTOFCUSTOMSANDTAXESASWELLASOTHERIMMUNITIESANDPRIVILEGESANDANYOTHERMILITA\
RYANDTECHNICALQUESTIONSCONCERNINGTHEORGANIZATIONANDOPERATIONOFTHEHEADQUARTERSMENTIONEDABOVESHALLBEDETERMI\
NEDBYASPECIALCONVENTIONWHICHSHALLCOMEINTOFORCENOTLATERTHANTHETREATYOFALLIANCE"

cipher = 'EOEYEGTRNPSECEHHETYHSNGNDDDDETOCRAERAEMHTECSEUSIARWKDRIRNYARANUEYICNTTCEIETUS'

Markov(reference,cipher)
```

# Markov( ): double 和 triple 是 dictionary，count( )會計算兩兩一組和三個一組的數量並回傳，dimension( )負責算出 7*11 或是 11*7，length 和 group_num 則是代表一排有幾個字母(11)和有幾排(7)，classify( )會將 cipher 分類成七個 group，接著跑三個 for 迴圈，第一層(i)代表一開始有幾個直排要排(group_num-2 是因為前兩行排好了)，第二層(j)代表有幾個直排尚未排好(從 i+2 開始，因為 i 代表幾行排好了，+2 是因為前兩行一開始就排好了)，第三層(k)代表在第幾行，每次取一直排，算出每行的條件機率，並找出擁有最大機率和的直排，swap( )可以交換其位置，使其依照順序排進去，最後再依照排好的順序印出結果



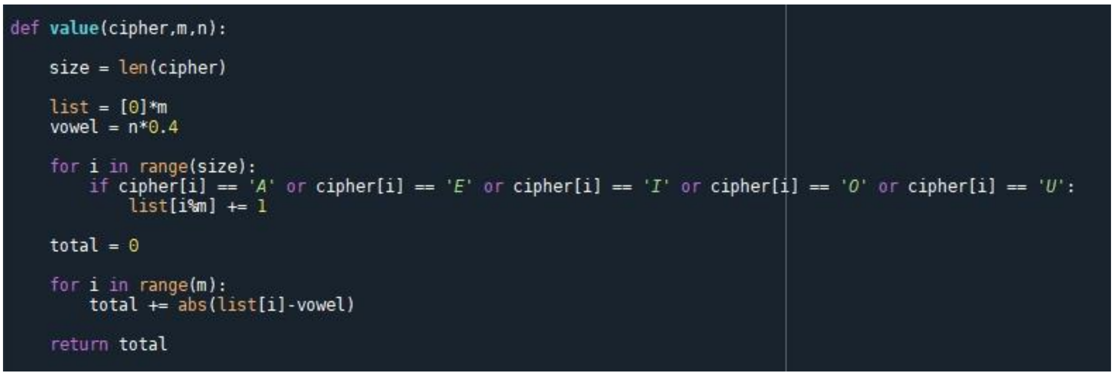
```
def Markov(reference,cipher):

    double, triple = count(reference)

    length,group_num = dimension(cipher,7,11)

    group = classify(cipher,group_num,length)

    for i in range(group_num-2):
        max_prob = 0
        for j in range(i+2,group_num):
            prob = 0
            for k in range(length):
                dou = ""
                dou += group[i][k]
                dou += group[i+1][k]
                tri = ""
                tri += group[i][k]
                tri += group[i+1][k]
                tri += group[j][k]
                if dou in double and tri in triple:
                    prob += triple[tri]/double[dou]
            if prob > max_prob:
                max_prob = prob
                max_index = j
        swap(group,max_index,i+2)

    for i in range(length):
        for j in range(group_num):
            print(group[j][i],end = '')
```

# count( ): 負責算出 reference 兩兩一組和三個一組的數量



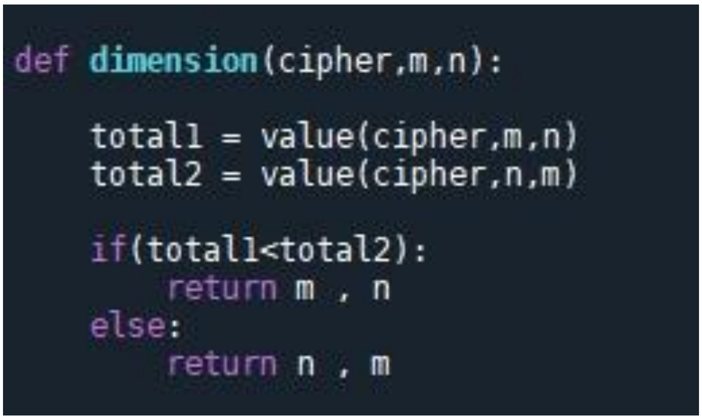
```
def count(reference):
    size = len(reference)
    double = {}
    triple = {}

    for i in range(size-2):
        string = ""
        string += reference[i]
        string += reference[i+1]

        if string in double:
            double[string] += 1
        else:
            double[string] = 1

        string += reference[i+2]

        if string in double:
            triple[string] += 1
        else:
            triple[string] = 1

    return double, triple
```

# value( ): 依照傳入的 m,n，去計算每一行該有的母音數量和實際母音數量差，(英文的母音數大約是 40%，所以乘上 0.4)，再將值回傳至 dimension( )



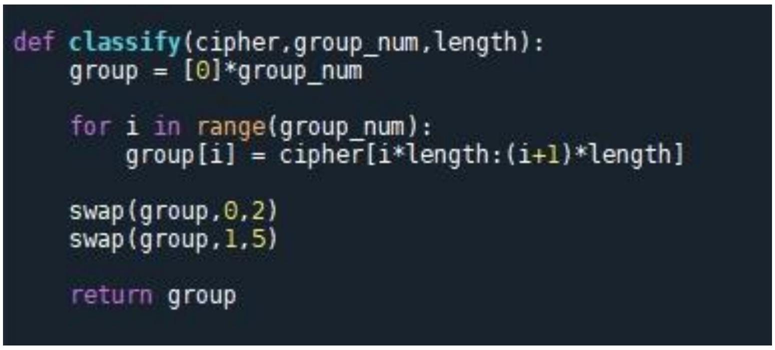
```
def value(cipher,m,n):

    size = len(cipher)

    list = [0]*m
    vowel = n*0.4

    for i in range(size):
        if cipher[i] == 'A' or cipher[i] == 'E' or cipher[i] == 'I' or cipher[i] == 'O' or cipher[i] == 'U':
            list[i%m] += 1

    total = 0

    for i in range(m):
        total += abs(list[i]-vowel)

    return total
```

# dimension( ): 比較(m,n)和(n,m)兩者的值，推出其 dimension



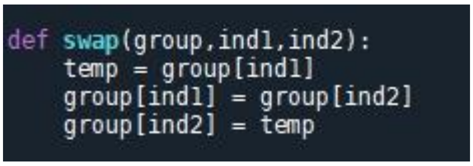
```
def dimension(cipher,m,n):

    total1 = value(cipher,m,n)
    total2 = value(cipher,n,m)

    if(total1<total2):
        return m , n
    else:
        return n , m
```

# classify( ): 將 cipher 分成七個 group，利用 swap( )將第一行是 G 和 R 的換至 group[0]和 group[1]，



```
def classify(cipher,group_num,length):
    group = [0]*group_num

    for i in range(group_num):
        group[i] = cipher[i*length:(i+1)*length]

    swap(group,0,2)
    swap(group,1,5)

    return group
```

# swap( ): 交換兩個 group



```
def swap(group,ind1,ind2):
    temp = group[ind1]
    group[ind1] = group[ind2]
    group[ind2] = temp
```

<u>Plaintext</u>

GREECEANNOUNCEDYESTERDAYITHADREACHEDAGREEMENTWITHTURKEYTOENDTHE
CYPRUSCRISISNS