

Python: without numpy or sklearn

Q1: Given two matrices please print the product of those two matrices

```
Ex 1: A  = [[1 3 4]
            [2 5 7]
            [5 9 6]]
      B  = [[1 0 0]
            [0 1 0]
            [0 0 1]]
      A*B = [[1 3 4]
            [2 5 7]
            [5 9 6]]
```

```
Ex 2: A  = [[1 2]
            [3 4]]
      B  = [[1 2 3 4 5]
            [5 6 7 8 9]]
      A*B = [[11 14 17 20 23]
            [23 30 36 42 51]]
```

```
Ex 3: A  = [[1 2]
            [3 4]]
      B  = [[1 4]
            [5 6]
            [7 8]
            [9 6]]
      A*B =Not possible
```

In [33]:

```

#https://towardsdatascience.com/a-complete-beginners-guide-to-matrix-multiplication-for-dat
#https://www.kite.com/python/answers/how-to-find-the-length-of-a-2d-array-in-python#:~:text
# write your python code here
# you can take the above example as sample input for your program to test
# it should work for any general input try not to hard code for only given input examples

# you can free to change all these codes/structure
# here A and B are List of Lists
def matrix_mul(Matrix_A,Matrix_B):
    # write your code

    r1=len(Matrix_A)
    c1=len(Matrix_A[0])
    r2=len(Matrix_B)
    c2=len(Matrix_B[0])
    if c1==r2:
        resultant_matrix=[[0 for i in range(r1)]for j in range(c2)]
        for r in range(r1):
            for c in range(c1):
                for c_2 in range(c2):
                    resultant_matrix[r][c]+=Matrix_A[r][c_2]*Matrix_B[c_2][c]

    else:
        return "Not Possible"
    return resultant_matrix
Matrix_A=[[1,3,4],[2,5,7],[5,9,6]]
Matrix_B=[[1,0,0],[0,1,0],[0,0,1]]

matrix_ans = matrix_mul(Matrix_A, Matrix_B)
print(matrix_ans)

```

```
[[1, 3, 4], [2, 5, 7], [5, 9, 6]]
```

Q2: Select a number randomly with probability proportional to its magnitude from the given array of n elements

consider an experiment, selecting an element from the list A randomly with probability proportional to its magnitude. assume we are doing the same experiment for 100 times with replacement, in each experiment you will print a number that is selected randomly from A.

Ex 1: A = [0 5 27 6 13 28 100 45 10 79]

let f(x) denote the number of times x getting selected in 100 experiments.

f(100) > f(79) > f(45) > f(28) > f(27) > f(13) > f(10) > f(6) > f(5) > f(0)

In [12]:

```

#https://www.sanfoundry.com/python-program-cumulative-sum-list/#:~:text=Python%20Program%20
#lecture video in this course proportional sampling
#https://www.geeksforgeeks.org/python-number-uniform-method/
import random
# write your python code here
# you can take the above example as sample input for your program to test
# it should work for any general input try not to hard code for only given input examples
# you can free to change all these codes/structure
# your code here for picking an element from with the probability propotional to its magnit
A = [0,5,27,6,13,28,100,45,10,79]
def pick_a_number_from_list(A):
    S=0
    for i in A:
        S+=i
    #print(S)
    idash=[]
    for i in A:
        idash.append(i/S)
    #print(idash)
    d=[sum(idash[0:i+1]) for i in range(0,len(idash))]
    #d
    r=random.uniform(0.0,1.0)

    c=[]
    for i in range(0,len(d)):
        if (r <= d[i]):
            return A[i]
prop_samp = []
for i in range(1,100):
    prop_samp.append(pick_a_number_from_list(A))
A=sorted(A)
for i in A:
    print(i,prop_samp.count(i))

```

```

0 0
5 0
6 3
10 5
13 5
27 7
28 9
45 16
79 23
100 31

```

Q3: Replace the digits in the string with

consider a string that will have digits in that, we need to remove all the not digits and replace the digits with #

Ex 1: A = 234	Output: ###
Ex 2: A = a2b3c4	Output: ###
Ex 3: A = abc	Output: (empty string)
Ex 5: A = #2a\$#b%c%561#	Output: #####

In [19]:

```
#w3school:regular Expression
import re
# write your python code here
# you can take the above example as sample input for your program to test
# it should work for any general input try not to hard code for only given input examples

# you can free to change all these codes/structure
# String: it will be the input to your program
def replace_digits(String):
    # write your code
    x= re.findall('\d',String)
    y=''.join(x)

    return(re.sub('\d','#',y)) # modified string which is after replacing the # with digits

A=['234','a2b3c4','abc','#2a$#b%c%561#']
for i in A:
    ans=replace_digits(i)
    print(ans)
```

```
###
###
```

```
#####
```

Q4: Students marks dashboard

consider the marks list of class students given two lists

Students =

```
['student1','student2','student3','student4','student5','student6','student7','student8','student9','student10']
```

Marks = [45, 78, 12, 14, 48, 43, 45, 98, 35, 80]

from the above two lists the Student[0] got Marks[0], Student[1] got Marks[1] and so on

your task is to print the name of students **a. Who got top 5 ranks, in the descending order of marks**

b. Who got least 5 ranks, in the increasing order of marks

d. Who got marks between >25th percentile <75th percentile, in the increasing order of marks

Ex 1:

```
Students=['student1','student2','student3','student4','student5','student6','student7','student8','student9','student10']
```

```
Marks = [45, 78, 12, 14, 48, 43, 47, 98, 35, 80]
```

a.

```
student8 98
```

```
student10 80
```

```
student2 78
```

```
student5 48
```

```
student7 47
```

b.

```
student3 12
```

```
student4 14
```

```
student9 35
```

```
student6 43
```

```
student1 45
```

c.

```
student9 35
```

```
student6 43
```

```
student1 45
```

```
student7 47
```

```
student5 48
```

In [5]:

```

##https://www.kite.com/python/answers/how-to-sort-a-list-of-tuples-by-the-second-value-in-p
# write your python code here
# you can take the above example as sample input for your program to test
# it should work for any general input try not to hard code for only given input examples

# you can free to change all these codes/structure
def display_dash_board(students, marks):
    students_within_25_and_75=[]
    result=list(zip(students,marks))
    top_5_students=sorted(result,key=lambda x:x[1],reverse=True)[:5]
    least_5_students=sorted(result,key=lambda x:x[1],reverse=False)[:5]
    students_within_25_and_75_test=sorted(result,key=lambda x:x[1],reverse=False)
    median=len(marks)/2
    IQR25=round(median/2)
    IQR75=round(median)+round(median/2)
    for i in range(IQR25,IQR75):
        students_within_25_and_75.append(students_within_25_and_75_test[i])

    return top_5_students, least_5_students, students_within_25_and_75
students=['student1','student2','student3','student4','student5','student6','student7','stu
marks = [45, 78, 12, 14, 48, 43, 47, 98, 35, 80]
top_5_students, least_5_students, students_within_25_and_75 = display_dash_board(students,
print(top_5_students,'\n', least_5_students, '\n',students_within_25_and_75 )

```

```

[('student8', 98), ('student10', 80), ('student2', 78), ('student5', 48),
('student7', 47)]
[('student3', 12), ('student4', 14), ('student9', 35), ('student6', 43),
('student1', 45)]
[('student9', 35), ('student6', 43), ('student1', 45), ('student7', 47),
('student5', 48)]

```

Q5: Find the closest points

consider you have given n data points in the form of list of tuples like $S=[(x_1,y_1),(x_2,y_2),(x_3,y_3),(x_4,y_4), (x_5,y_5),\dots,(x_n,y_n)]$ and a point $P=(p,q)$

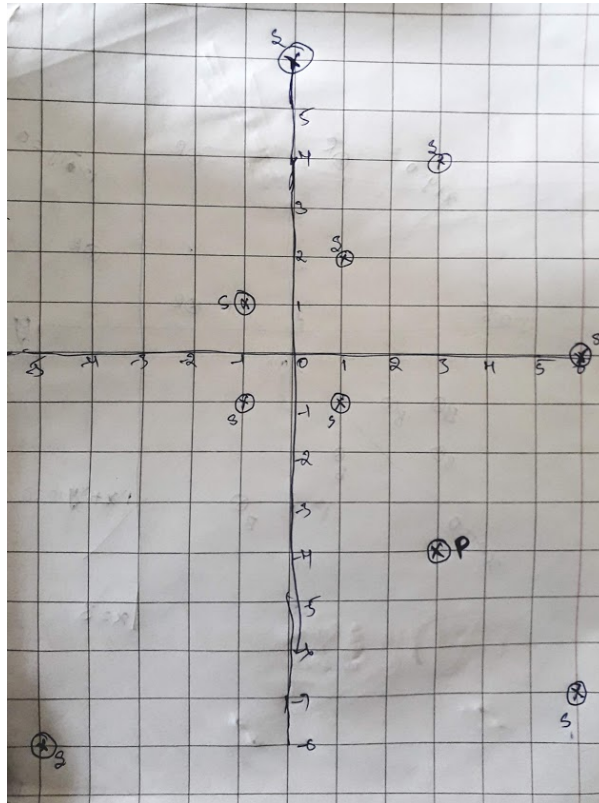
your task is to find 5 closest points(based on cosine distance) in S from P

cosine distance between two points (x,y) and (p,q) is defined as $\cos^{-1}\left(\frac{(x \cdot p + y \cdot q)}{\sqrt{(x^2 + y^2)} \cdot \sqrt{(p^2 + q^2)}}\right)$

Ex:

$S = [(1,2), (3,4), (-1,1), (6,-7), (0,6), (-5,-8), (-1,-1), (6,0), (1,-1)]$

$P = (3,-4)$



Output:

(6, -7)

(1, -1)

(6, 0)

(-5, -8)

(-1, -1)

In [10]:

```

#https://www.kite.com/python/answers/how-to-get-the-first-element-of-each-tuple-in-a-list-i
#https://www.w3schools.com/python/ref_math_acos.asp
#https://www.w3schools.com/sql/func_mysql_acos.asp
import math

# write your python code here
# you can take the above example as sample input for your program to test
# it should work for any general input try not to hard code for only given input examples
# you can free to change all these codes/structure

# here S is list of tuples and P is a tuple of len=2
def closest_points_to_p(S, P):
    cos_distance=[]
    x=[]
    for i in S:
        dist1=i[0]*P[0]+i[1]*P[1]
        sq_dist=math.sqrt((i[0]**2)*(P[0]**2))+math.sqrt((i[1]**2)*(P[1]**2))
        if sq_dist !=0:
            cos_distance = math.acos(dist1/sq_dist)
            x.append(cos_distance)

    zip_v = list(zip(S,x))
    #print(zip_v)
    Y = [S for S in sorted(zip_v, key=lambda i:i[1])][:5]
    #print(Y)
    return Y

S = [(1,2),(3,4),(-1,1),(6,-7),(0, 6),(-5,-8),(-1,-1),(6,0),(1,-1)]
P= (3,-4)
points = closest_points_to_p(S, P)
for i, j in points:
    print(i)

```

```

(6, -7)
(6, 0)
(1, -1)
(-5, -8)
(-1, -1)

```

Q6: Find Which line separates oranges and apples

consider you have given two set of data points in the form of list of tuples like

```

Red = [(R11,R12),(R21,R22),(R31,R32),(R41,R42),(R51,R52),...,(Rn1,Rn2)]
Blue=[(B11,B12),(B21,B22),(B31,B32),(B41,B42),(B51,B52),...,(Bm1,Bm2)]

```

and set of line equations(in the string formate, i.e list of strings)

```
Lines = [a1x+b1y+c1,a2x+b2y+c2,a3x+b3y+c3,a4x+b4y+c4,...,K lines]
```

Note: you need to string parsing here and get the coefficients of x,y and intercept

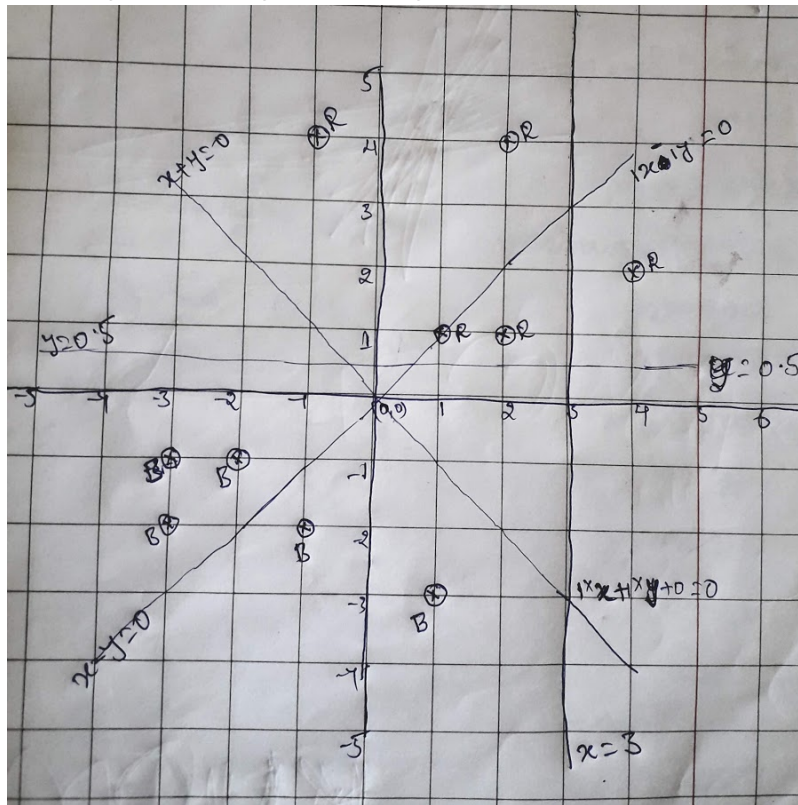
your task is to for each line that is given print "YES"/"NO", you will print yes, if all the red points are one side of the line and blue points are other side of the line, otherwise no

Ex:

Red= [(1,1),(2,1),(4,2),(2,4), (-1,4)]

Blue= [(-2,-1),(-1,-2),(-3,-2),(-3,-1),(1,-3)]

Lines=["1x+1y+0", "1x-1y+0", "1x+0y-3", "0x+1y-0.5"]



Output:

YES

NO

NO

YES

In [11]:

```

#https://stackoverflow.com/questions/1841565/valueerror-invalid-literal-for-int-with-base-1
#https://www.geeksforgeeks.org/python-check-if-all-elements-in-list-follow-a-condition/
#https://realpython.com/python-eval-function/#:~:text=Python's%20eval()%20allows%20you,or%2
#https://note.nkmk.me/en/python-str-replace-translate-re-sub/
import math
def i_am_the_one(red,blue,line):
    red_ls=[]
    blue_ls=[]
    #print(red,blue,line)
    for j in Red:
        red = eval(line.replace('x',('*'+str(j[0]))).replace('y',('*'+str(j[1]))))
        red_ls.append(red)
        res_red = all(float(ele) > 0 for ele in red_ls)
    for j in Blue:
        blue = eval(line.replace('x',('*'+str(j[0]))).replace('y',('*'+str(j[1]))))
        blue_ls.append(blue)
        res_blue = all(float(ele) > 0 for ele in blue_ls)
    #print(res_red,res_blue)
    if res_red==1 and res_blue==0:
        return "yes"
    else:
        return "no"

Red= [(1,1),(2,1),(4,2),(2,4), (-1,4)]
Blue= [(-2,-1),(-1,-2),(-3,-2),(-3,-1),(1,-3)]
Lines=["1x+1y+0", "1x-1y+0", "1x+0y-3", "0x+1y-0.5"]

for i in Lines:
    yes_or_no = i_am_the_one(Red, Blue, i)
    print(yes_or_no)

```

```

yes
no
no
yes

```

Q7: Filling the missing values in the specified formate

You will be given a string with digits and '_' (missing value) symbols you have to replace the '_' symbols as explained

Ex 1: `_, _, _, 24 ==> 24/4, 24/4, 24/4, 24/4` i.e. we. have distributed the 24 equally to all 4 places

Ex 2: `40, _, _, _, 60 ==> (60+40)/5, (60+40)/5, (60+40)/5, (60+40)/5, (60+40)/5 ==> 20, 20, 20, 20, 20` i.e. the sum of (60+40) is distributed equally to all 5 places

Ex 3: `80, _, _, _, _ ==> 80/5, 80/5, 80/5, 80/5, 80/5 ==> 16, 16, 16, 16, 16` i.e. the 80 is distributed equally to all 5 missing values that are right to it

Ex 4: `_, _, 30, _, _, _, 50, _, _`

`==>` we will fill the missing values from left to right

a. first we will distribute the 30 to left two missing values (`10, 10, 10, _, _, 50, _, _`)

b. now distribute the sum (10+50) missing values in between (`10, 10, 12, 12, 12, 12, 12, _, _`)

c. now we will distribute 12 to right side missing values (`10, 10, 12, 12, 12, 12, 4, 4, 4`)

for a given string with comma separate values, which will have both missing values numbers like ex: `"_, _, x, _, _, "` you need fill the missing values

Q: your program reads a string like ex: `"_, _, x, _, _, "` and returns the filled sequence

Ex:

Input1: `"_, _, _, 24"`

Output1: `6, 6, 6, 6`

Input2: `"40, _, _, _, 60"`

Output2: `20, 20, 20, 20, 20`

Input3: `"80, _, _, _, _"`

Output3: `16, 16, 16, 16, 16`

Input4: `"_, _, 30, _, _, _, 50, _, _"`

Output4: `10, 10, 12, 12, 12, 12, 4, 4, 4`

In [13]:

```

#https://stackoverflow.com/questions/7889820/how-to-obtain-the-last-index-of-a-list:
#https://stackoverflow.com/questions/2582138/finding-and-replacing-elements-in-a-list
# write your python code here
# you can take the above example as sample input for your program to test
# it should work for any general input try not to hard code for only given input strings
# you can free to change all these codes/structure
def curve_smoothing(string):

    spl_ls = S.split(',')
    orig_ls=[]

    for i in range(len(spl_ls)) :
        if spl_ls[i] != '_' or i==len(spl_ls)-1:
            orig_ls.append(i)

    first = 0
    initial_sum =0

    for i in orig_ls:#passing the position[2,6,8]
        if spl_ls[i]!='_':
            initial_sum +=int(spl_ls[i]) #first and next value sum
        else:
            initial_sum=smoothing_sum #updating to the latest sum

    end = i
    no_pos = (end-first)+1
    smoothing_sum = initial_sum/no_pos
    #print(smoothing_sum)
    initial_sum=smoothing_sum
    for x in range(len(spl_ls)):
        if first<=x<=end:
            spl_ls[x]=smoothing_sum
        else:
            spl_ls[x]

    first = i

    return spl_ls

S= "_,_,30,_,_,50,_,_"
smoothed_values= curve_smoothing(S)
print(smoothed_values)

```

```
[10.0, 10.0, 12.0, 12.0, 12.0, 12.0, 4.0, 4.0, 4.0]
```

Q8: Filling the missing values in the specified formate

You will be given a list of lists, each sublist will be of length 2 i.e. $[[x,y],[p,q],[l,m]..[r,s]]$ consider its like a matrix of n rows and two columns 1. the first column F will contain only 5 uniques values (F_1, F_2, F_3, F_4, F_5) 2. the second column S will contain only 3 uniques values (S_1, S_2, S_3)

your task is to find

- Probability of $P(F=F_1|S==S_1)$, $P(F=F_1|S==S_2)$, $P(F=F_1|S==S_3)$
- Probability of $P(F=F_2|S==S_1)$, $P(F=F_2|S==S_2)$, $P(F=F_2|S==S_3)$
- Probability of $P(F=F_3|S==S_1)$, $P(F=F_3|S==S_2)$, $P(F=F_3|S==S_3)$
- Probability of $P(F=F_4|S==S_1)$, $P(F=F_4|S==S_2)$, $P(F=F_4|S==S_3)$
- Probability of $P(F=F_5|S==S_1)$, $P(F=F_5|S==S_2)$, $P(F=F_5|S==S_3)$

Ex:

$[[F1, S1], [F2, S2], [F3, S3], [F1, S2], [F2, S3], [F3, S2], [F2, S1], [F4, S1], [F4, S3], [F5, S1]]$

- $P(F=F1|S==S1)=1/4, P(F=F1|S==S2)=1/3, P(F=F1|S==S3)=0/3$
- $P(F=F2|S==S1)=1/4, P(F=F2|S==S2)=1/3, P(F=F2|S==S3)=1/3$
- $P(F=F3|S==S1)=0/4, P(F=F3|S==S2)=1/3, P(F=F3|S==S3)=1/3$
- $P(F=F4|S==S1)=1/4, P(F=F4|S==S2)=0/3, P(F=F4|S==S3)=1/3$
- $P(F=F5|S==S1)=1/4, P(F=F5|S==S2)=0/3, P(F=F5|S==S3)=0/3$

In [41]:

```
#https://stackoverflow.com/questions/1388818/how-can-i-compare-two-lists-in-python-and-retu
#https://www.geeksforgeeks.org/python-count-occurrences-element-list/
# write your python code here
# you can take the above example as sample input for your program to test
# it should work for any general input try not to hard code for only given input strings
# you can free to change all these codes/structure
def compute_conditional_probabilites(F,S):
    FS_count=0
    S_count=0
    for i in range(len(A)):
        if(A[i][1]==S):#storing S value and count and comparing with A
            S_count=S_count+1
            if(A[i][0]==F):#storing FS count
                FS_count=FS_count+1
    prob_score = FS_count/S_count
    print('P(F==',F,'|S==',S,'=',FS_count,'/',S_count,' Prob_score',prob_score)

A = [['F1', 'S1'], ['F2', 'S2'], ['F3', 'S3'], ['F1', 'S2'], ['F2', 'S3'], ['F3', 'S2'], ['F2', 'S1'], ['F4', 'S1'], ['F4', 'S3'], ['F5', 'S1']]
# #[F1, S1],[F1,S2],[F1 ,S3],[F2 ,S1],[F2 ,S2],[F2 ,S3],[F3 ,S1],[F3 ,S2],[F3 ,S3],[F4 ,S1]
#genrating for each F,all 3 S values combination
for F in ['F1', 'F2', 'F3', 'F4', 'F5']:
    for S in ['S1', 'S2', 'S3']:
        compute_conditional_probabilites(F,S)
# print the output as per the instructions
```

```
P(F== F1 |S== S1 = 1 / 4 Prob_score 0.25
P(F== F1 |S== S2 = 1 / 3 Prob_score 0.3333333333333333
P(F== F1 |S== S3 = 0 / 3 Prob_score 0.0
P(F== F2 |S== S1 = 1 / 4 Prob_score 0.25
P(F== F2 |S== S2 = 1 / 3 Prob_score 0.3333333333333333
P(F== F2 |S== S3 = 1 / 3 Prob_score 0.3333333333333333
P(F== F3 |S== S1 = 0 / 4 Prob_score 0.0
P(F== F3 |S== S2 = 1 / 3 Prob_score 0.3333333333333333
P(F== F3 |S== S3 = 1 / 3 Prob_score 0.3333333333333333
P(F== F4 |S== S1 = 1 / 4 Prob_score 0.25
P(F== F4 |S== S2 = 0 / 3 Prob_score 0.0
P(F== F4 |S== S3 = 1 / 3 Prob_score 0.3333333333333333
P(F== F5 |S== S1 = 1 / 4 Prob_score 0.25
P(F== F5 |S== S2 = 0 / 3 Prob_score 0.0
P(F== F5 |S== S3 = 0 / 3 Prob_score 0.0
```

Q9: Given two sentences S1, S2

You will be given two sentences S1, S2 your task is to find

- Number of common words between S1, S2
- Words in S1 but not in S2
- Words in S2 but not in S1

Ex:

S1= "the first column F will contain only 5 uniques values"

S2= "the second column S will contain only 3 uniques values"

Output:

- 7
- ['first', 'F', '5']
- ['second', 'S', '3']

In [30]:

```
#https://stackoverflow.com/questions/4260280/if-else-in-a-list-comprehension
# write your python code here
# you can take the above example as sample input for your program to test
# it should work for any general input try not to hard code for only given input strings
# you can free to change all these codes/structure
def string_features(S1, S2):
    # your code
    list1=S1.split()
    list2=S2.split()
    count=0
    for l1 in list1:
        if l1 in list2 :
            count=count+1
    l1_match=[l1 for l1 in list1 if l1 not in list2]
    l2_match=[l2 for l2 in list2 if l2 not in list1]
    return count, l1_match, l2_match

S1= "the first column F will contain only 5 uniques values"
S2= "the second column S will contain only 3 uniques values"
a,b,c = string_features(S1, S2)
print(a,b,c)
```

```
7 ['first', 'F', '5'] ['second', 'S', '3']
```

Q10: Given two sentences S1, S2

You will be given a list of lists, each sublist will be of length 2 i.e. [[x,y],[p,q],[l,m]..[r,s]] consider its like a martrix of n rows and two columns

- the first column Y will contain interger values
- the second column Y_{score} will be having float values

Your task is to find the value of

$f(Y, Y_{score}) = -1 * \frac{1}{n} \sum_{foreach Y, Y_{score} pair} (Y \log_{10}(Y_{score}) + (1 - Y) \log_{10}(1 - Y_{score}))$ here n is the number of rows in the matrix

Ex:

```
[[1, 0.4], [0, 0.5], [0, 0.9], [0, 0.3], [0, 0.6], [1, 0.1], [1, 0.9], [1, 0.8]]
```

output:

0.4243099

$$\frac{-1}{8} \cdot ((1 \cdot \log_{10}(0.4) + 0 \cdot \log_{10}(0.6)) + (0 \cdot \log_{10}(0.5) + 1 \cdot \log_{10}(0.5)) + \dots + (1 \cdot \log_{10}(0.8) + 0 \cdot \log_{10}(0.3)))$$

In [32]:

```
#https://www.w3schools.com/python/ref_math_log10.asp
import math
# write your python code here
# you can take the above example as sample input for your program to test
# it should work for any general input try not to hard code for only given input strings
# you can free to change all these codes/structure
def compute_log_loss(A):
    # your code

    y_yscore=A
    sum=0
    for i in range(len(y_yscore)):
        Y_pred=y_yscore[i][1]
        Y_actual=y_yscore[i][0]
        sum=sum+(Y_actual*math.log10(Y_pred)+(1-Y_actual)*math.log10(1-Y_pred))
        logloss=-1*sum/len(y_yscore)

    return logloss

A = [[1, 0.4], [0, 0.5], [0, 0.9], [0, 0.3], [0, 0.6], [1, 0.1], [1, 0.9], [1, 0.8]]
loss = compute_log_loss(A)
print(loss)
```

0.42430993457031635