Q3

You have to write a Python program which does the following:

* ***takes*** a **list** as input
* ***converts*** the list into a **Pandas Series**
* ***reshapes*** the Series into a **dataframe**of the order **m**x **n**, i.e. the dataframe has **m**rows and **n** columns

**Input Format**

* The first line contains two space-separated integers, representing the values of **m** and **n** respectively.
* The second line contains space-separated integers, which represent the elements of the **list**

**Constraints**

* 2 <= **m, n** <= 10
* The product of **m**and **n**should be equal to the **length**of the input **list**

**Output Format**

* If the product of **m** and **n** equals the length of the input**list**, print the converted **dataframe**
* If the product of **m** and **n** is**NOT EQUAL**to the length of the input**list**, print the message "**Weird Order!**"

import numpy as np

import pandas as pd

inp=pd.Series(input().split())

m=int(inp[0])

n=int(inp[1])

input\_list=pd.Series(input().split())

if m\*n == len(input\_list):

print(pd.DataFrame(input\_list.values.reshape(m,n)))

else:

print("Weird Order!")

Q4

**Problem Statement**

Write a program that does the following:

* ***takes*** three space-separated **integers** as input
* ***generates*** a series of evenly spaced numbers as another **list**using the above values (using **NumPy arange**)
* ***takes*** another **list** as input
* ***creates*** two **Pandas Series** out of the lists
* ***merges*** both Series along the **index**axis, to form a **Data Frame**
* ***takes*** **column names** as inputs
* ***prints*** some (**n**) rows from the first OR last of the Data Frame along with the **column names**, depending on the input

**NOTE:**

* This DataFrame should contain the **column names** as well as the data values, printed out to the console (STDOUT)

**Input Format**

* The first line contains three space-separated **integers:**
* **stp**- the starting interval for the generated list
* **enp** - the ending interval for the generated list
* **step** - the step value (spacing between values) for the elements of the list
* **e.g.**For the values 2, 12 and 3, the generated list **list1** will be:
* [2, 5, 8, 11]
* The second line contains the elements of the second list **list2**
* The third line contains two space-separated **strings** which should be the ***column names*** for the first and second lists respectively
* The last line contains two space-separated **integers**:
* **n** - This represents the number of rows that you have to print from the DataFrame.
* **pos** - This represents the position from where the rows have to be extracted. This can be only **1** (first) or **0** (last).

**Output Format**

* Depending on the values of **n** and **pos**, print the first/last **n** rows of the DataFrame on to the output console (STDOUT)

**NOTE:**

* When printing the output, the elements of **list1** should always be present in the first column and elements of **list2** should always be present in the second column, else your test-cases will fail.

**Evaluation Parameters**

**Sample Input**

2 12 3

axe bar cat div eli fif gor had

NUM VAL

6 1

**Sample Output**

NUM VAL

0 2.0 axe

1 5.0 bar

2 8.0 cat

3 11.0 div

4 NaN eli

5 NaN fif

**Explanation**

* There are only 4 elements in **list1** while there are 8 elements in **list2**
* With the values of **n**(6) and **pos**(1), the **1st 6** rows of the DataFrame are printed.
* Since, there are less number of corresponding elements in **list1**, hence the rest of the mapping under the column **NUM**shows up as **NaN**
* The column names, obtained from input, are **NUM**and **VAL**. They have been used as column names and can be seen at the head of the output.

import pandas as pd

import numpy as np

num=input().split()

stp =int(num[0])

enp =int(num[1])

step =int(num[2])

list2 = input().split()

colname=input().split()

col1=colname[0]

col2=colname[1]

num1=input().split()

n = int(num1[0])

pos = int(num1[1])

# Write the logic to perform operations

list1 = np.arange(stp,enp,step)

x=pd.Series(list1)

y=pd.Series(list2)

#data={col1:x,col2:y}

df = pd.concat([x,y],axis=1)

df.rename(columns={0:col1,1:col2},inplace=True)

#df = pd.DataFrame(list(zip(list1, list2)), columns =[col1, col2])

if pos==1:

print(df.head(n))

elif pos==0:

print(df.tail(n))

Q5

**Problem Statement**

You are working as a Data Analyst for a digital marketing firm. The firm has recently launched various marketing campaigns for its clients. As part of the campaigns, the firm has delivered a certain number of marketing cold emails to the customer.

You are given some data related to the name of the client as the **campaign name**, the **number of emails** delivered as part of the campaign and the **email open count** for each campaign.

Your task here is to calculate the **open rate** for each campaign and print the output in the form of Boolean values for the open rates as follows:

* If the open rate for a campaign is greater than the mean open rate the output for that campaign should be “True”
* If the open rate for a campaign is lower than the mean open rate the output for that campaign should be “False”

Use **pandas**function and create a DataFrame from Dict of ndarrays / Lists

**Input​ ​Format**

* The first line contains an integer **T**, denoting the number of elements in the dictionary
* The second line contains **T**space-separated values containing data related to**Campaign Name**.
* The third line contains **T**space-separated values containing data related to **Number of Emails**.
* The fourth line contains **T**space-separated values containing data related to **Email Open Rate**.

**Output​ ​Format**

* Print the values as True/False as per the conditions provided in the problem statement

**Evaluation Parameters**

**Sample​ ​Input**

4

SoftBank Apple Microsoft Amazon

2845 3425 2942 4215

1253 3256 2100 1252

**Sample​ ​Output**

False

True

True

False

import pandas as pd

t=int(input())

l2=input().split()

capName=[str (x) for x in l2]

l3=input().split()

mail=[int (x) for x in l3]

l4=input().split()

openrate=[int (x) for x in l4]

dict1={i:[j,k] for i,j,k in zip(capName,mail,openrate)}

df=pd.DataFrame(dict1)

mean=df.iloc[1].mean()

for i in df.iloc[1]:

if i>mean:

print("True")

else:

print("False")

Q6 Write a Python function that takes a string as input and returns a dictionary where the keys are alphabets from the strings and values for those keys are count of those alphabets in the string.

def AlphabetCount(s):

d={}#create a dictionary

for i in s:

keys=d.keys()

if i in keys:

d[i]=d[i]+1

else:

d[i]=1

return d #return dictionary

#read the string S

s=input()

print(AlphabetCount(s))#Call the function

#Display the dictionary

Q7 word freq

def WordCount(s):

d=dict()#create a dictionary

w=s.split()

for i in w:

if i in d:

d[i]=d[i]+1

else:

d[i]=1

return d

#your code here

#return dictionary object

#read a string S

s=input()

print(WordCount(s))

#Call the function

#Display the dictionary object

Q8 sum of multiples

def SumOfMultiples(n):

s=0

for i in range (3,n+1):

if (i%3==0 or i%5==0):

s=s+i

return s

#return the value of sum

#Read the value of limit

n=int(input())

print(SumOfMultiples(n))

#call the function

# display sum of multiples of 3 and 5 within the given limit

#### Q9 Write a Python function to move every zero to the right side of a given list of integers.

# Read the variable from STDIN

n = int(input())

l=[]

for i in range(0,n):

a=int(input())

l.append(a)

length=len(l)

c=0

for i in range(length):

if l[i]!=0:

l[c]=l[i]

c=c+1

while c<length:

l[c]=0

c=c+1

print(l)

Q10 A positive integer m is a sum of squares if it can be written as k + p where k > 0, p > 0 and both k and p are perfect squares.

Write a Python function sumofsquares(m) that takes an integer m returns True if m is a sum of squares and False otherwise. (If m is not positive, your function should return False.)

The number 41 can be written as sum of two perfect squares 16 and 25.

The number 30 cannot be written as sum of two perfect squares.

import math

def sumofsquares(m):

s=math.sqrt(m)

f=math.floor(s)

for i in range (1,f+1):

for j in range (f,0,-1):

if i\*\*2+j\*\*2==m:

return True

return False

#read m

m=int(input())

print(sumofsquares(m))

Q Armstrong

n=int(input())

p=n

l=0

s=0

while p>0:

l=l+1

p=p//10

p=n

while p>0:

r=p%10

s=s+(r\*\*3)

p//=10

if (s==n and l==3):

print ("ARMSTRONG")

elif l!=3:

print ("INVALID\_INPUT")

else:

print ("NOT ARMSTRONG")

Q

#function to put positive numbers in the beginning and negative numbers at the last

numlist=[]#list of integers

pos=[]#list of positive integers

neg=[]#list of negative integers

j = 0

# Take the Input From the User

n = int(input())

for i in range(0,n):

a= int(input())

numlist.append(a)

while(j < n):

if(numlist[j] >= 0):#if the integer is positive

pos.append(numlist[j])#add the integer to the list of positive integers

else:

neg.append(numlist[j])#add the integer to the list of negative integers

j = j + 1

print(pos+neg)#merge both the lists