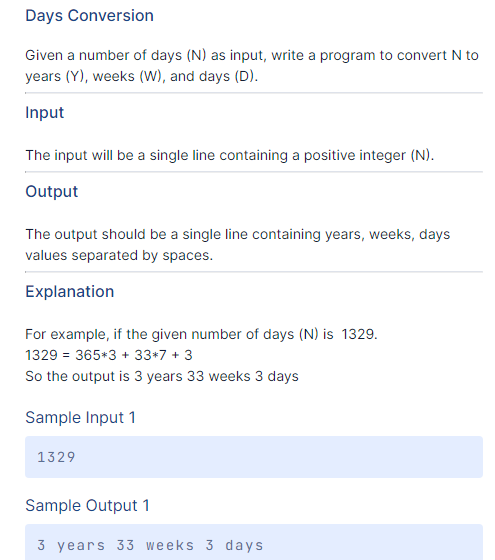
Grand Assignment 1

Days Conversation



n=int(input())

year=n/365

weeks=(n%365)/7

days=(n%365)%7

year\_value=int(year)

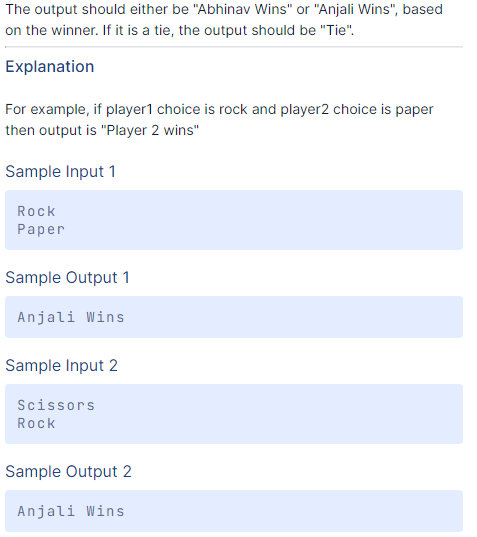
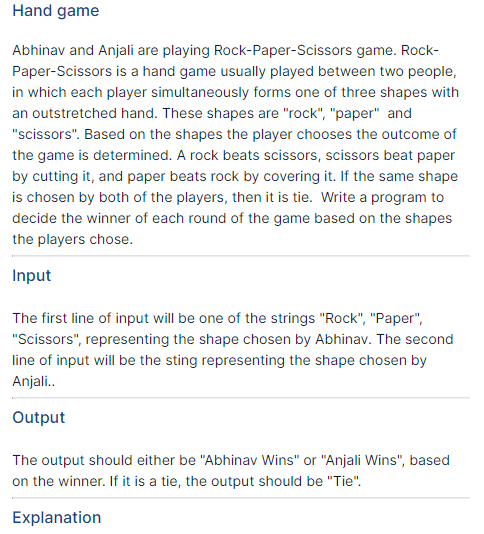
Weeks\_value=int(weeks)

days\_value=int(days)

result=(str(year\_value)+" "+"years "+str(Weeks\_value)+" "+"weeks "+str(days\_value)+" "+"days ")

print(result)

Hand Game



anjali=input()

abhi=input()

if abhi==anjali:

print("Tie")

elif abhi=="Rock":

if anjali=="Paper":

print("Abhinav Wins")

else:

print("Anjali Wins")

elif abhi=="Paper":

if anjali=="Scissors":

print("Abhinav Wins")

else:

print("Anjali Wins")

elif abhi=="Scissors":

if anjali=="Rock":

print("Abhinav Wins")

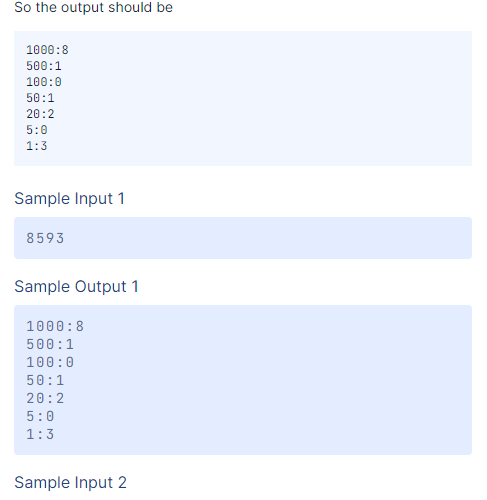
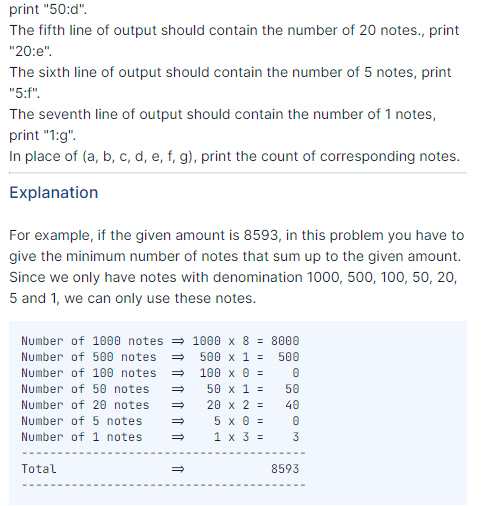
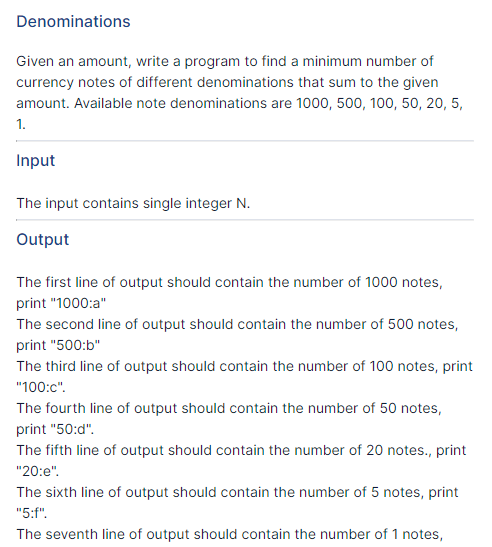
else:

print("Anjali Wins")

else:

print("Not Valid Play")

Denomination



number=int(input())

note1000=0

note500=0

note100=0

note50=0

note20=0

note5=0

note1=0

if number>=1000:

note1000=int(number/1000)

number=(number%1000)

if number>=500:

note500=int(number/500)

number=(number%500)

if number>=100:

note100=int(number/100)

number=(number%100)

if number>=50:

note50=int(number/50)

number=(number%50)

if number>=20:

note20=int(number/20)

number=(number%20)

if number>=5:

note5=int(number/5)

number=(number%5)

note1=number

print("1000:" + str(note1000))

print("500:" + str(note500))

print("100:" + str(note100))

print("50:" + str(note50))

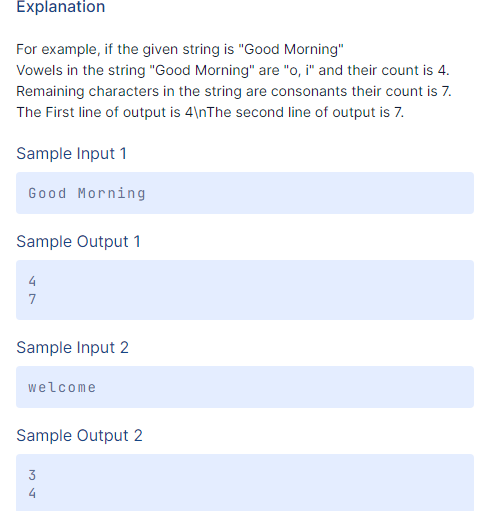
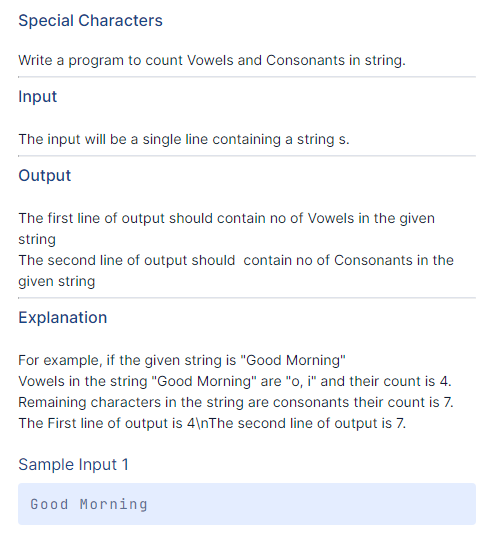
print("20:" + str(note20))

print("5:" + str(note5))

print("1:" + str(note1))

Grand Assignment 2

Special Character



word=input()

vowels=("a","e","i","o","u")

vowels\_count=0

string=0

for i in word:

if i==" ":

continue

if (i in vowels):

vowels\_count=vowels\_count+1

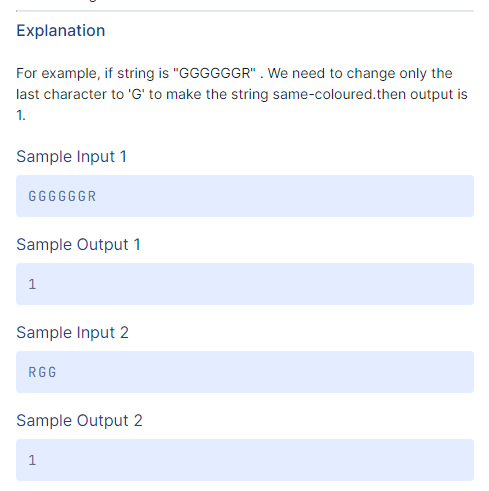
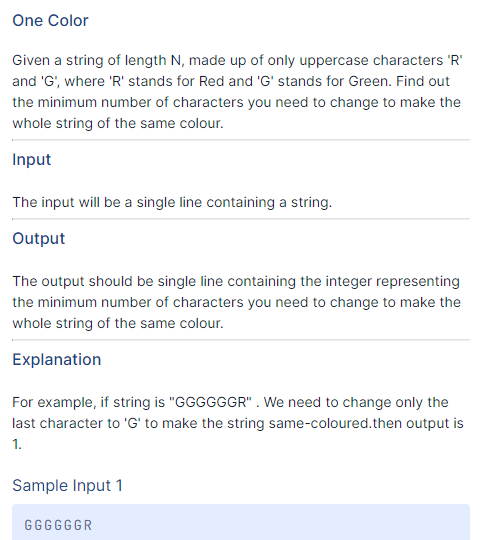
elif(i not in vowels):

string=string+1

print(vowels\_count)

print(string)

One Color



word=input()

value=0

n1=word.count("R")

n2=word.count("G")

if n1>n2:

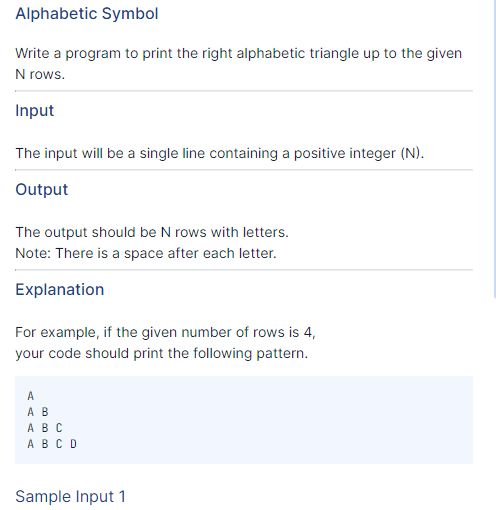
value=n2

else:

value=n1

print(value)

Alphabetic Number



n=int(input())

data=""

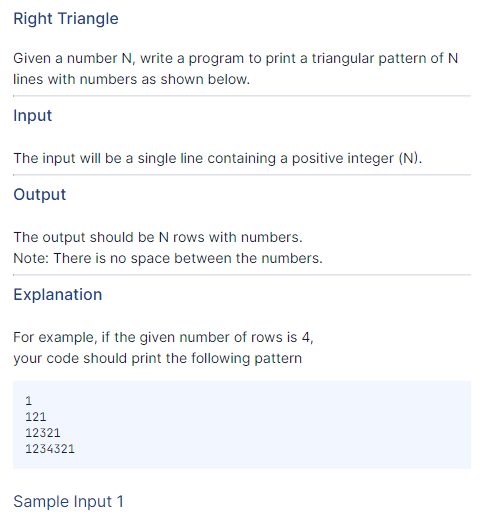
for i in range(1,n+1):

value=64+i

data=data+chr(value)+" "

print(data)

Right Triangle



m=int(input())

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

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

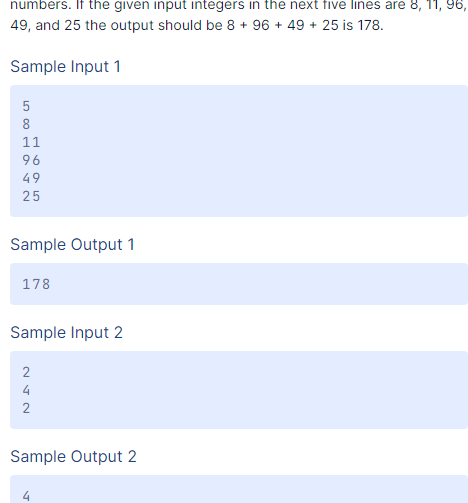
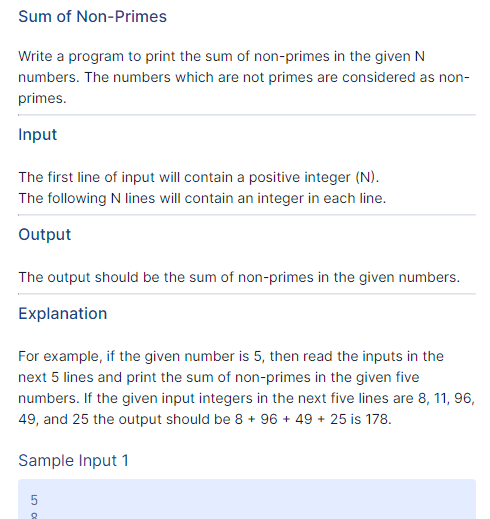
print(j,end="")

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

print(j,end="")

print()

Sum of Non-Prime



n = int(input())

is\_primes = []

for count in range(n):

read = int(input())

is\_primes.append(read)

non\_primes = []

for num in is\_primes:

for i in range(2, num):

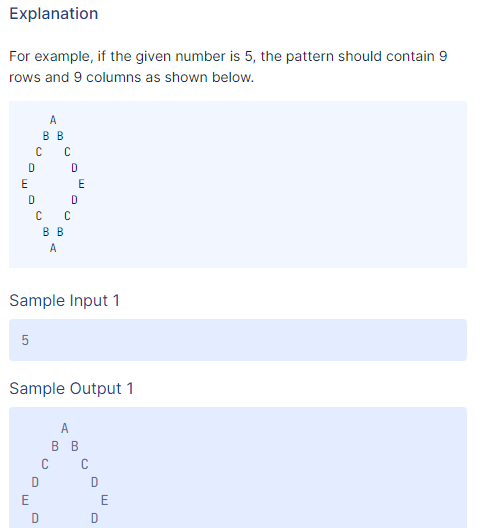
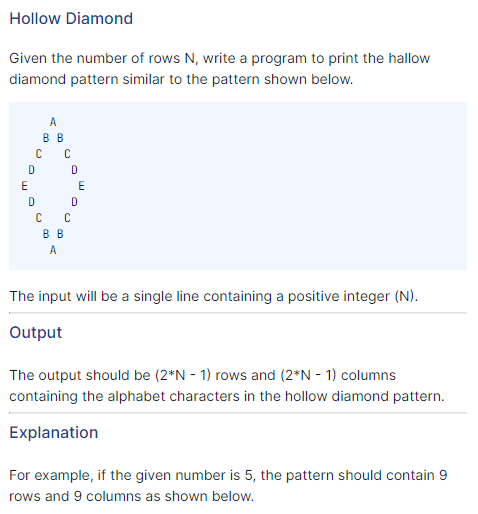
if num % i == 0:

non\_primes.append(num)

break

print(sum(non\_primes))

Hollow Diamond



n = int(input())

left=n-1

mid=-1

char="A"

print(' '\*left,char,sep="")

for \_ in range(n-1):

left=left-1

mid=mid+2

char=chr(ord(char)+1)

print(' '\*left,char,' '\*mid,char,sep="")

for \_ in range(n-2):

left=left+1

mid=mid-2

char=chr(ord(char)-1)

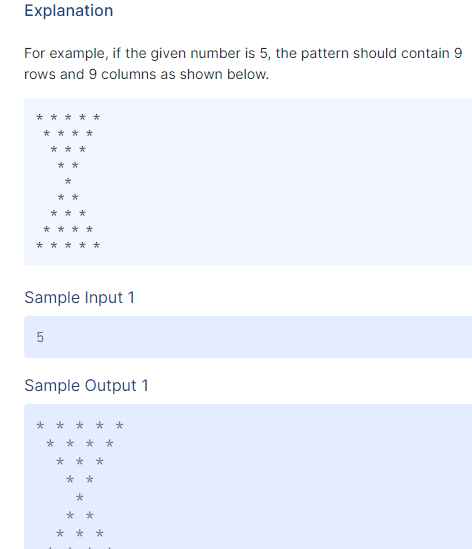
print(' '\*left,char,' '\*mid,char,sep="")

left=left+1

char="A"

print(' '\*left,char,sep="")

Sandglass Star



row = int(input())

for i in range(row):

for j in range(i):

print(" ", end="")

for j in range(row-i):

print("\* ", end="")

print()

for i in range(1,row):

for j in range(row-i-1):

print(" ", end="")

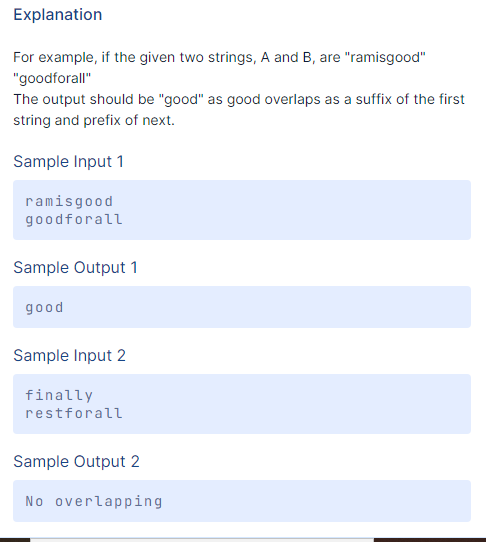
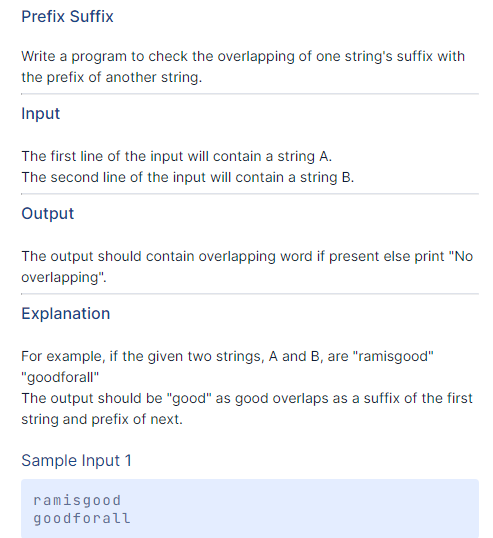
for j in range(i+1):

print("\* ", end="")

print()

Grand assignment 3

Prefix Sufix



m=input()

n=input()

prefix=0

for i in range(len(m)):

if m[i:i+2]==n[0:2]:

prefix=prefix+1

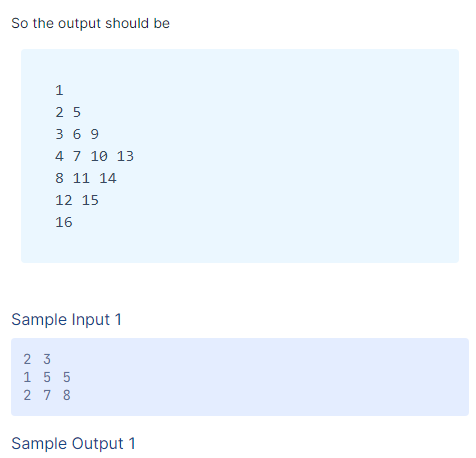
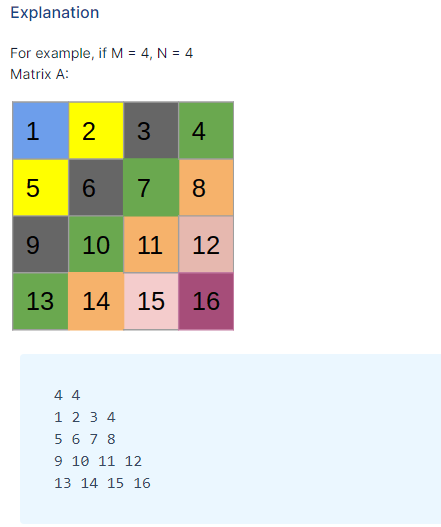
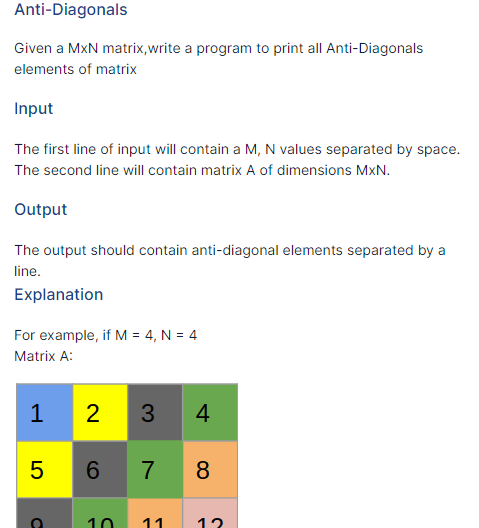
print(m[i:])

break

if prefix==0:

print("No overlapping")

Anti Diagonal



def anti(m,n,matrix):

anti=[[]for \_ in range(m+n-1)]

for i in range(m\*n):

index=(i//n)+(i%n)

anti[index].append(matrix[i])

for j in anti:

print(\*j)

m,n=list(map(int,input().split()))

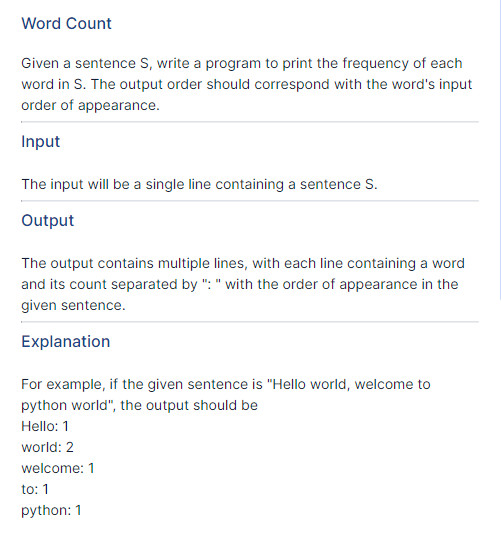
a=[]

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

a+=map(int,input().split())

anti(m,n,a)

Word Count



word= (input())

value = word.split()

count=[]

for i in value:

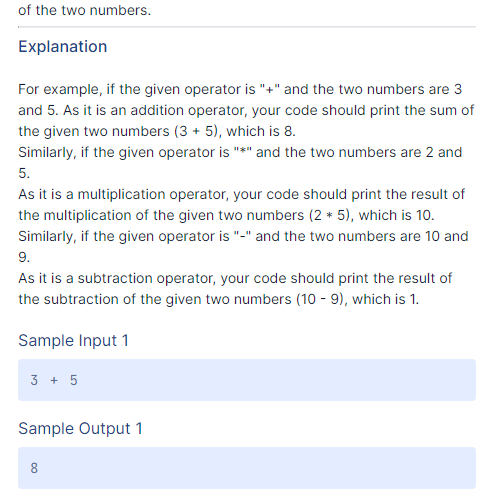
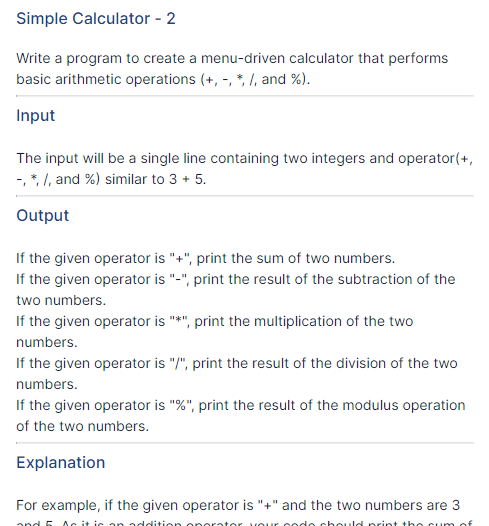
if i not in count:

count.append(i)

for i in range(0, len(count)):

print("%0s:%2s"%(count[i].ljust(1),value.count(count[i])))

Simple Calculator 2



num1,oprator,num2 = input().split()

if oprator == "+":

print(int(num1) + int(num2))

elif oprator == "-":

print(int(num1)-int(num2))

elif oprator == "\*":

print(int(num1)\*int(num2))

elif oprator == "/":

print(int(num1)/int(num2))

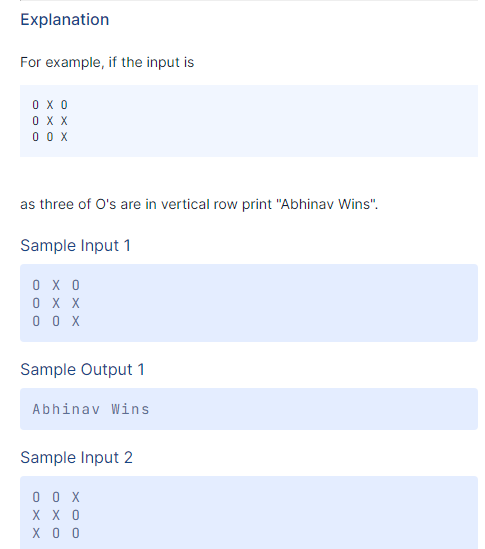
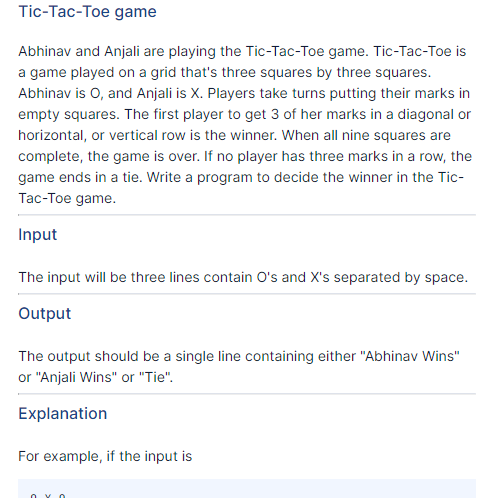
elif oprator == "%":

print(int(num1)%int(num2))

else:

print("Error")

Tic Tak Tao



O\_WINS = "Abhinav Wins"

X\_WINS = "Anjali Wins"

TIE = "Tie"

def find\_winner(matrix):

n = len(matrix)

sum\_main\_diag = 0

sum\_other\_diag = 0

for i in range(n):

sum\_current\_row = 0

sum\_current\_column = 0

for j in range(n):

sum\_current\_row += matrix[i][j]

sum\_current\_column += matrix[j][i]

if sum\_current\_row == n or sum\_current\_column == n:

return X\_WINS

if sum\_current\_row == 0 or sum\_current\_column == 0:

return O\_WINS

sum\_main\_diag += matrix[i][i]

sum\_other\_diag += matrix[i][n - 1 - i]

if sum\_main\_diag == n or sum\_other\_diag == n:

return X\_WINS

if sum\_main\_diag == 0 or sum\_other\_diag == 0:

return O\_WINS

return TIE

if \_\_name\_\_ == "\_\_main\_\_":

import sys

# read from standard input

source = sys.stdin

# read from a file called 'input.txt' (uncomment the next line)

# source = open('input.txt', 'r')

# reading the board state from the input source

input\_matrix = list(map(lambda line: line.split(), source.readlines()))

# converting the board to a binary form where '0' is 0 and 'X' is 1

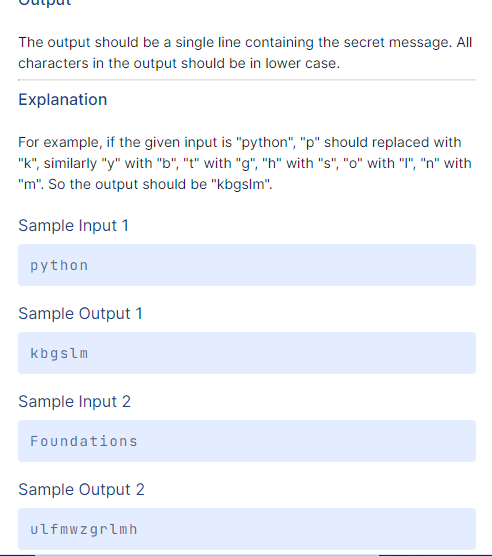
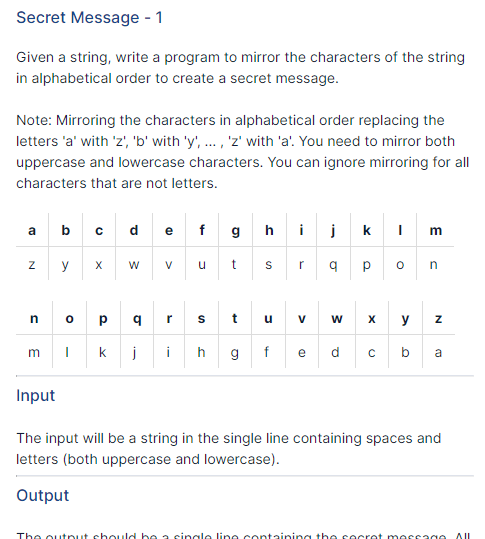
binary\_matrix = [list(map(lambda x: 1 if x == 'X' else 0, row)) for row in input\_matrix]

print(find\_winner(binary\_matrix))

source.close()

Grand Assignment 4

Serect Message1



alpha={"a":"z","b":"y","c":"x","d":"w","e":"v","f":"u","g":"t","h":"s","i":"r","j":"q",

"k":"p","l":"o","m":"n","n":"m","o":"l","p":"k","q":"j","r":"i","s":"h","t":"g","u":"f",

"v":"e","w":"d","x":"c","y":"b","z":"a"

}

text\_alpha="abcdefghijklmnopqrstuvwxyz"

text=input()

data=text.lower()

secrect\_text=""

for i in data:

if i in text\_alpha:

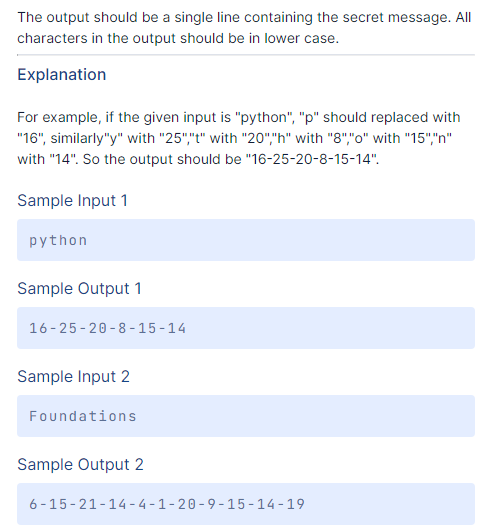
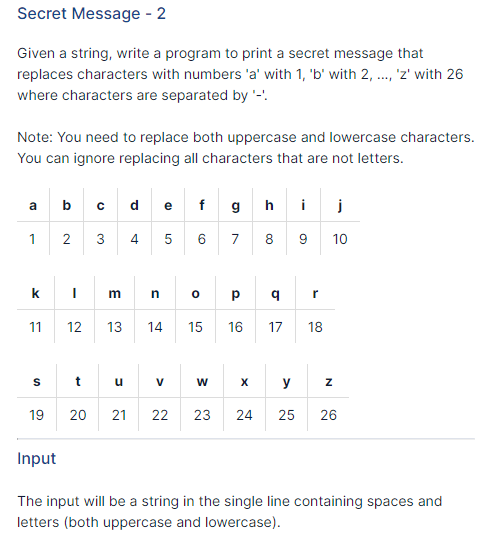
secrect\_text=secrect\_text+alpha[i]

else:

secrect\_text=secrect\_text+i

print(secrect\_text)

Secret Message 2



alphabet = "abcdefghijklmnopqrstuvwxyz"

cipher = dict()

for num, letter in enumerate(alphabet, 1):

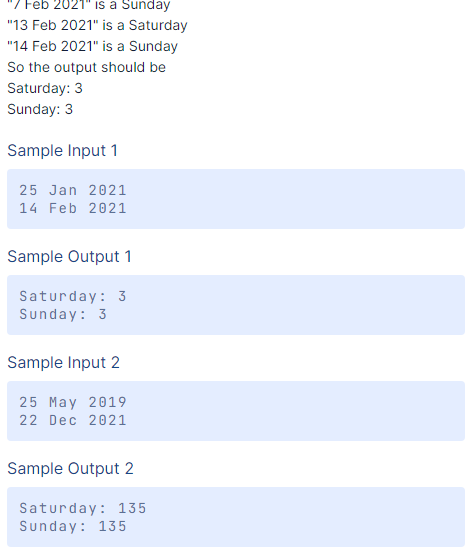
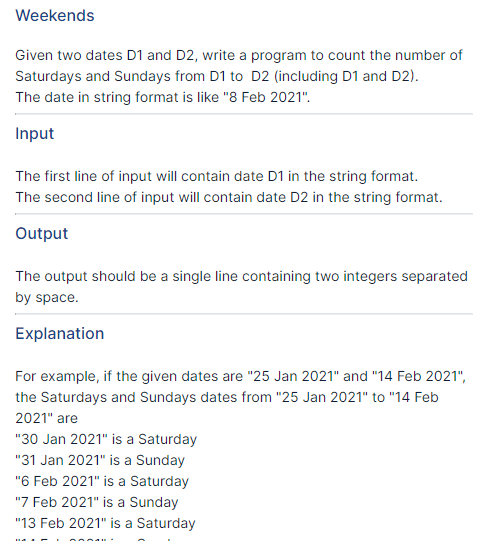
cipher[ord(letter)] = str(num) + "-"

s = input().lower().split()

for word in s :

print(word.translate(cipher)[:-1], end = " ")

Weekends



import datetime

date\_start\_str = input()

date\_end\_str = input()

date\_start = datetime.datetime.strptime(date\_start\_str, '%d %b %Y')

date\_end = datetime.datetime.strptime(date\_end\_str, '%d %b %Y')

day = datetime.timedelta(days=1)

count\_saturday = 0

count\_sunday = 0

while date\_start <= date\_end:

if date\_start.isoweekday() == 6:

count\_saturday += 1

if date\_start.isoweekday() == 7:

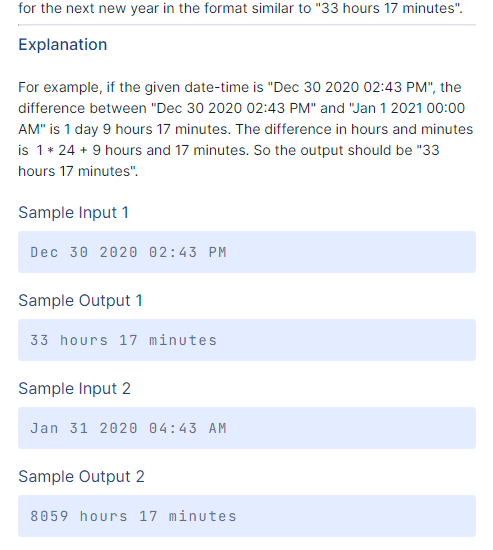
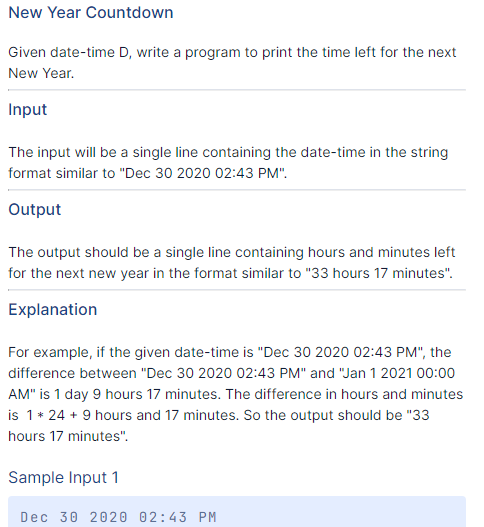
count\_sunday += 1

date\_start += day

print("{}: {}".format("Saturday",count\_saturday))

print("{}: {}".format("Sunday",count\_sunday))

New Year Countdown



from datetime import datetime, time

def date\_diff\_in\_seconds(dt2, dt1):

timedelta = dt2 - dt1

return timedelta.days \* 24 \* 3600 + timedelta.seconds

def dhms\_from\_seconds(seconds):

minutes, seconds = divmod(seconds, 60)

hours, minutes = divmod(minutes, 60)

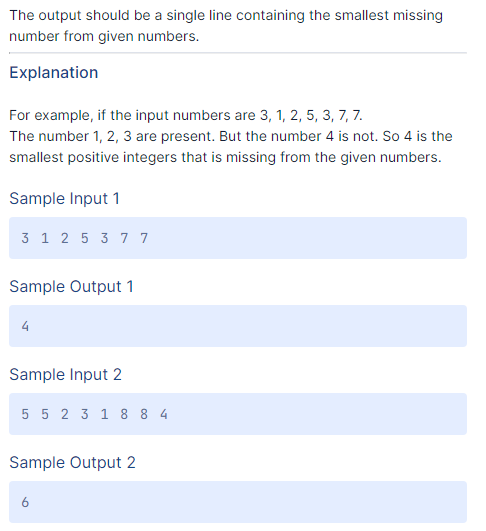
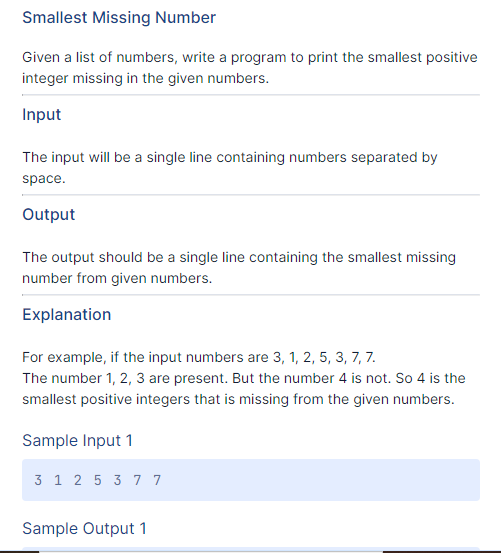
return (hours, minutes)

date1 = datetime.strptime(input(), "%b %d %Y %I:%M %p")

date2 = datetime.strptime("Jan 1 2021 00:00 AM", "%b %d %Y %H:%M %p")

print("%d hours %d minutes" %(dhms\_from\_seconds(date\_diff\_in\_seconds(date2, date1))))

Smallest Missing Number



numbers = input()

numberListStr = numbers.split(' ')

numberListInt = [int(number) for number in numberListStr]

smallestMissing = 1

while True:

if smallestMissing not in numberListInt:

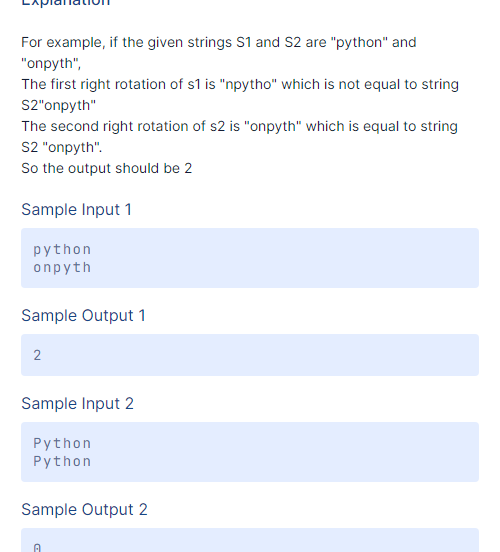
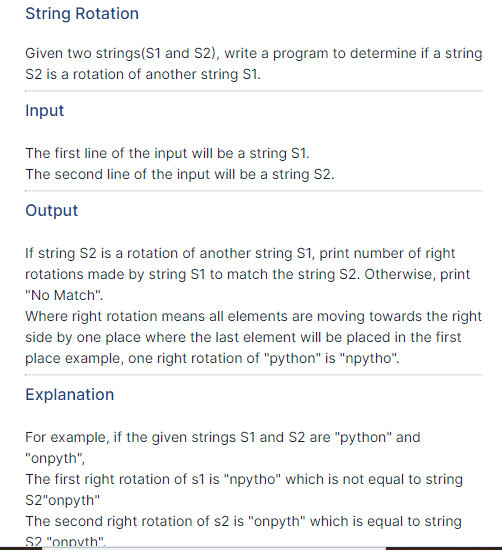
break

smallestMissing += 1

print(smallestMissing)

Grand Assignment 5

String Rotation



def main():

string1=input("")

string2=input("")

nRotation=getNumberRotation(string1,string2)

if nRotation==-1:

print("No Match")

else:

print(str(nRotation))

def getNumberRotation(string1,string2):

tempString=string1

nRotation =0

while tempString!=string2:

nRotation+=1

tempString = string1[len(string1) - nRotation:] + string1[0 : len(string1) - nRotation]

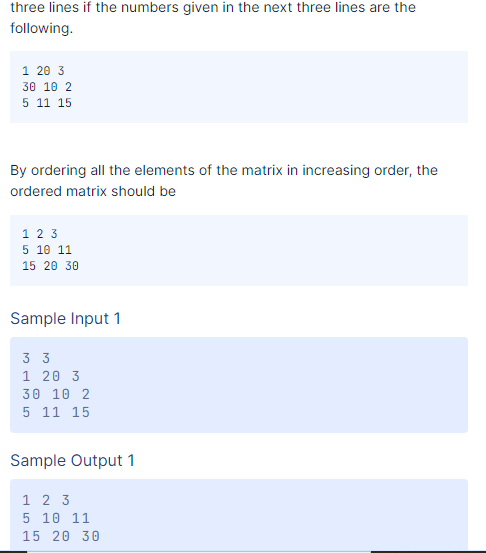
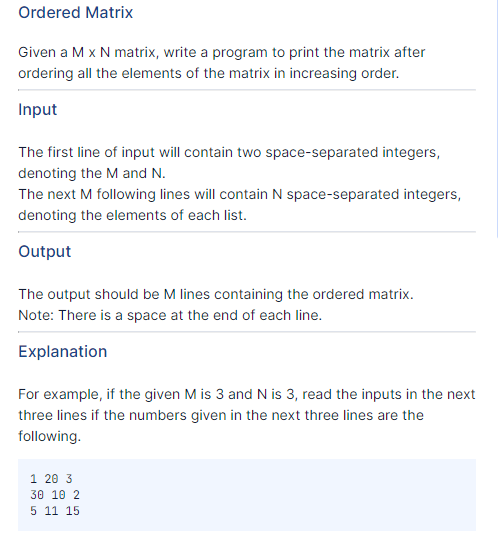
if nRotation>=len(string1):

return -1

return nRotation

main()

Order Matrix



m,n = map(int, (input().split()))

list\_matrix = []

list\_all\_numbers = []

for i in range(m):

list\_i = list(map(int, (input().split())))

list\_all\_numbers.extend(list\_i)

list\_matrix.append(list\_i)

list\_all\_numbers.sort()

for j in range(0,m):

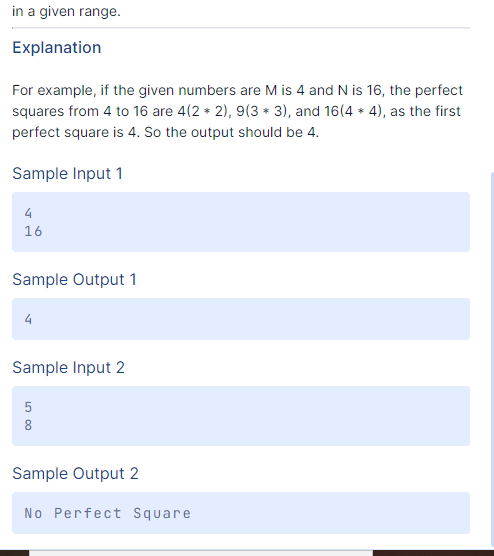
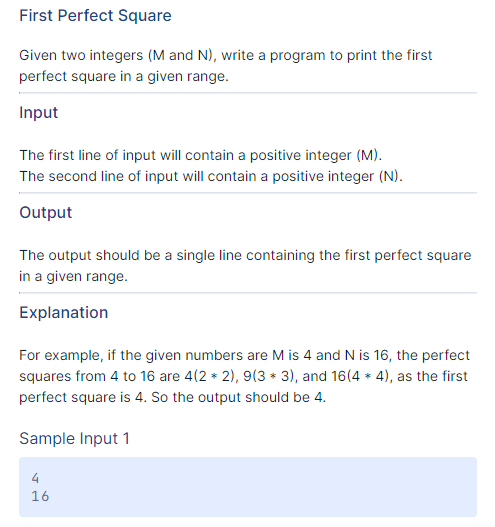
res = ""

for i in range(0,n):

res = res + str(list\_all\_numbers[(j\*n) + i]) + " "

print(res)

First Perfect Square



def Squares(a, b):

for i in range (a, b + 1):

j = 1;

while j \* j <= i:

if j \* j == i:

yield j\*j

j = j + 1

i = i + 1

a = int(input())

b = int(input())

x = [v for v in Squares(a,b)]

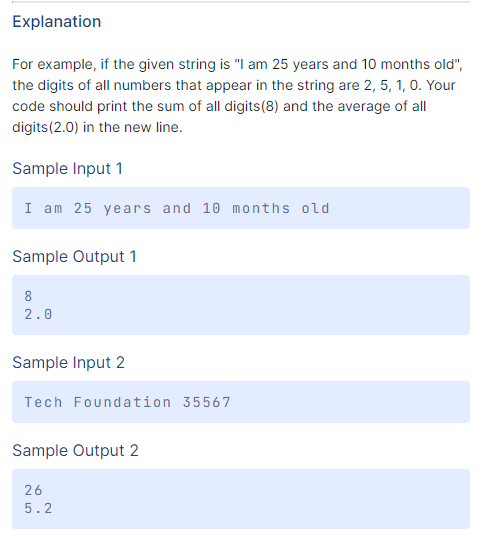
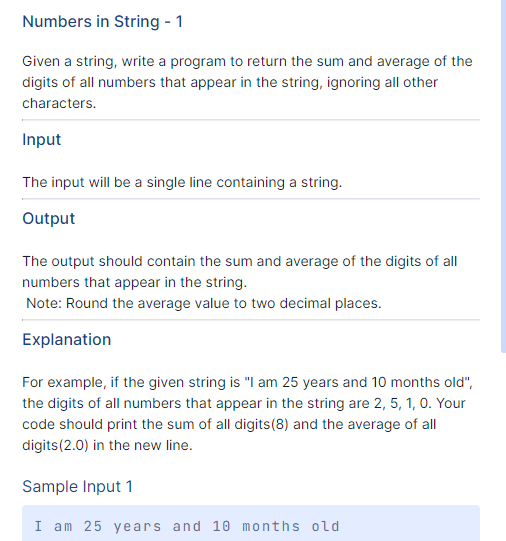
if x:

print(min(x))

else:

print("No Perfect Square")

Number in String 1



string=input()

integer=0

counter=0

for i in string:

if (i.isdigit()):

integer=integer+int(i)

counter=counter+1

else:

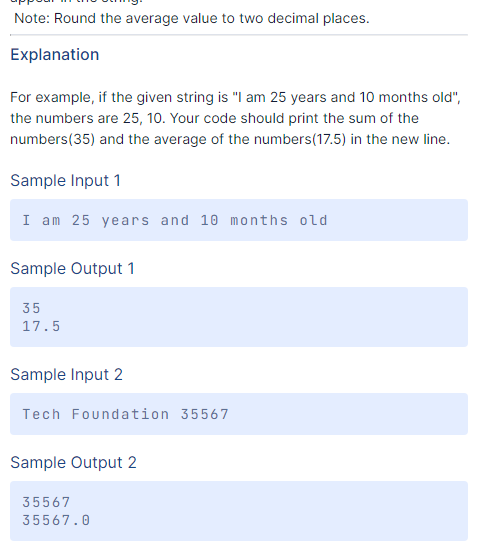
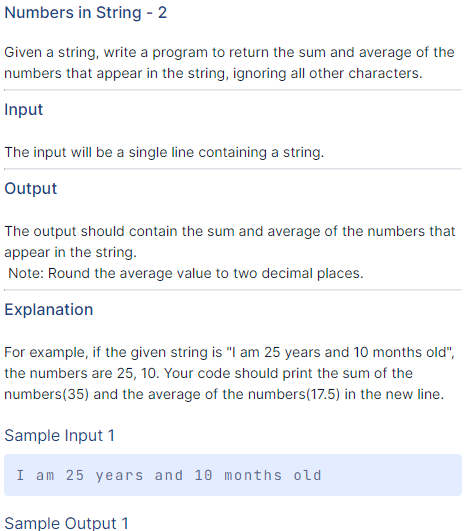
continue

average=integer/counter

print(integer)

print(round(average,2))

Number in String 2



n=input().split()

digits=""

count=0

for i in n:

p=""

get=""

for j in range(len(i)):

if i[j].isdigit():

if i[j-1].isdigit():

get+=i[j]

else:

get+=" "+i[j]

digits+=get+" "

amt=0

ran=len(digits.split())

for i in digits.split():

amt+=int(i)

print(amt)

average=round(amt/ran,2)

print(average)