Pentafox2 Interview questions

5. Time to solve some problems. (5 questions - 1 point each)

#Create a method/function to check whether the given 2 strings are anagram to one another or not.

Return true if they are anagram.

s1=input(' ')

s2=input(' ')

if(sorted(s1)==sorted(s2)):

print("True")

else:

print("False")

#Return the sum of the prime numbers between given range (including given numbers).

Eg., Input: 3 & 60. Output should print the sum of all the primes lying between 3 & 60

from math import sqrt

sum = 0

test = int(input())

max = int(input())

for x in range(test,max+1):

if x == 1:

pass

else:

half = int(sqrt(x)) + 1

for y in range(2,half):

res = x%y

if res==0:

break

else:

sum = sum + x

print(sum)

# Return the total number of days from the date given to till now.

Eg., Input: 24-01-1995 & Output should return the total days

import datetime

from datetime import date

def numOfDays(date1, today):

return (today-date1).days

date\_entry = input('Enter a date in YYYY-MM-DD format')

year, month, day = map(int, date\_entry.split('-'))

date1 = datetime.date(year, month, day)

today = date.today()

print(numOfDays(date1, today), "Total days")

#Create a simple encryption program.

- Eg., Input: Hi awesome! Output: H#9\_a#23#5#19#15#13#5! -

Replace each character with the correct alphabetical order number except the ?first character of each word.

Eg., Hi -> H#9 (add # before every number) - Replace spaces with \_

string=input()

word\_list=string.split(" ")

output=[]

for i in word\_list:

len\_string=len(i)

result=""

for j in range(len\_string):

if j==0:

result=i[j]

else:

ascii=ord(i[j])

if 65<=ascii and ascii<=90:

ascii=ascii-64

result=result+'#'+str(ascii)

elif 97<=ascii and ascii<=122:

ascii=ascii-96

result=result+'#'+str(ascii)

else:

result=result+chr(ascii)

output.append(result)

output.append("\_")

len\_string=len(output)

for i in range(len\_string-1):

print(output[i],end="")

#Convert array to object based on data type. - Eg., Input: ["test", "name", 45, "anything", true, 78] -

Eg., Output: { "strings": ["test", "name", "anything"], "numbers": [45, 78], "booleans": [true] }

class MyClass:

def \_\_init\_\_(self,output):

self.output = output

def return\_op(self):

return self.output

if \_\_name\_\_ == '\_\_main\_\_':

input= ["test","name",45,"anything",True,78]

boolean=[]

number=[]

string=[]

dicti={}

for i in input:

if type(i)==type(True):

boolean.append(i)

elif type(i)==type(1):

number.append(i)

else:

string.append(i)

obj=MyClass(string)

dicti["strings"]=obj.return\_op()

obj1=MyClass(number)

dicti["numbers"]=obj1.return\_op()

obj2=MyClass(boolean)

dicti["boolean"]=obj2.return\_op()

print(dicti)