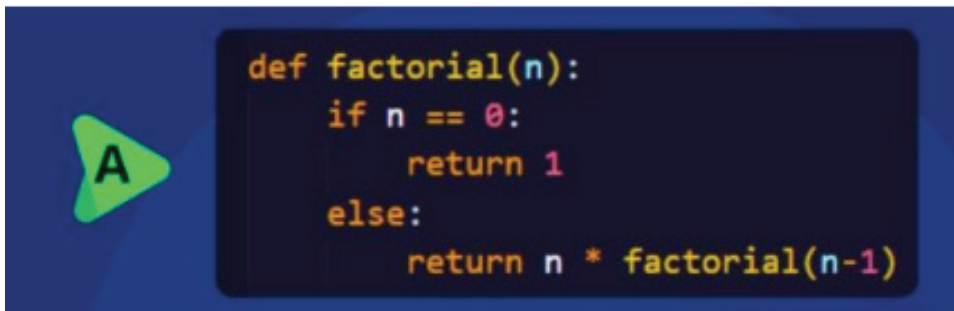




PROBLEMA

1. ¿Cual de las siguientes opciones contiene el código Python correcto para calcular el factorial de un



Este código implementa el cálculo del factorial utilizando recursión. Es correcto, ya que sigue la definición matemática del factorial:

$$n! = n \times (n - 1)!$$

PROBLEMA

2. CLAVE MURCIELAGO

Descripción:

La clave MURCIELAGO es una sencilla clave que como Boy Scout aprender debes. Es fácil de realizar por su ventaja de cambiar las letras por símbolos numéricos, trabajando con la siguiente tabla de conversión:

M=0, U=1, R=2, C=3, I=4, E=5, L=6, A=7, G=8, O=9

De manera que si nuestro texto original fuera BOLIVIA
el resultado de la codificación sería: B964V47

Entrada

La entrada consiste en un entero N que es el número de casos de prueba, seguido de las N líneas que contiene una cadena escrita totalmente en letras mayúsculas.

Salida

Imprimir N líneas con las cadenas convertidas

Ejemplo Entrada

3
BOLIVIA
UNIVERSITARIO
ENTRADA

Ejemplo Salida

B964V47
1N4V52S4T7249
5NT27D7



SOURCE IN PYTHON

```
# _____ Crear Archivo _____
def convertirClaveMurcielago(cadenas):
    conversion = {
        'M': '0', 'U': '1', 'R': '2', 'C': '3', 'I': '4',
        'E': '5', 'L': '6', 'A': '7', 'G': '8', 'O': '9'
    }
    resultados = []
    for cadena in cadenas:
        resultado = ""
        for letra in cadena:
            if letra in conversion:
                resultado += conversion[letra]
            else:
                resultado += letra
        resultados.append(resultado)
    return resultados

n = int(input("Ingrese el número de palabras: "))
cadenas = []

print('_____ Ingrese Solo PALABRAS Mayúsculas _____')
for i in range(1, n + 1):
    palabra = input(f'{i}= ').strip()
    cadenas.append(palabra)

resultados = convertirClaveMurcielago(cadenas)

print('_____ Mostrando Resultados _____')
for resultado in resultados:
    print(resultado)
```

DESK CHECK

```
Ingrese el número PALABRAS: 3
_____ Ingrese Solo PALABRAS Mayusculas _____
1=BOLIVIA
2=UNIVERSITARIO
3=ENTRADA
_____ Mostrando Resultados _____
B964V47
1N4V52S4T7249
5NT27D7
```





PROBLEMA

- 3 Realiza un programa que pida 10 números por teclado y que los almacene en un array. A continuación se mostrará el contenido de ese array junto al índice (0 - 9) utilizando para ello una tabla. Seguidamente el programa pasará los primos a las primeras posiciones, desplazando el resto de números (los que no son primos) de tal forma que no se pierda ninguno. Al final se debe mostrar el array resultante.

SOURCE IN PYTHON

```
def leerTamanho():
    n=int(input('Ingrese tamanho='))
    return n

def llenarVector(n):
    V=[0]*n
    for i in range(n):
        V[i]=int(input(f'[{i}]:'))
    return V

def mostrarVector(n,V):
    print('\n+_____Elementos del Vector_____')
    print('[',end='')
    for i in range(n):
        print(V[i],end=', ')
    print(V[i],end=', ')
    print(']')

def veriPrimo(numero):
    if numero < 2:
        return False
    c=0
    for i in range(1, numero+1):
        if numero % i == 0:
            c=c+1
    if c==2:
        return True
    else:
        return False

def ordenarPrimos(n, V):
    primo=[]
    noPrimo=[]
    for i in range(n):
        if veriPrimo(V[i])==True:
            primo.append(V[i])
        else:
            noPrimo.append(V[i])
    Arrayfina= primo+noPrimo
    return Arrayfina

n=leerTamanho()
vector=llenarVector(n)
mostrarVector(n,vector)
vectorOrdenado =ordenarPrimos(n, vector)
mostrarVector(n,vectorOrdenado)
```

Índice	0	1	2	3	4	5	6	7	8	9
Valor	20	5	7	4	32	9	2	14	11	6

Array inicial

Índice	0	1	2	3	4	5	6	7	8	9
Valor	5	7	2	11	20	4	32	9	14	6

DESK CHECK

Ingrese tamanho=10

[0]:20

[1]:5

[2]:7

[3]:4

[4]:32

[5]:9

[6]:2

[7]:14

[8]:11

[9]:6

+_____Elementos del Vector_____

[20 , 5 , 7 , 4 , 32 , 9 , 2 , 14 , 11 , 6 ,]

+_____Elementos del Vector_____

[5 , 7 , 2 , 11 , 20 , 4 , 32 , 9 , 14 , 6 ,]





PROBLEMA

4. En un archivo almacenar el nombre, apellido, y 3 notas de materias

SOURCE IN PYTHON

import os

def crearArchivo():

#voy crear carpeta Data

if not os.path.exists('data'):

os.makedirs('data')

print('Carpeta "data" creada.')

if os.path.exists('data/alumnos.txt'):

print('El archivo "alumnos.txt" \n ya existe en la carpeta "data".')

else:

print('El archivo "alumnos.txt" no existe\n en la carpeta "data". Se creará.')

archivo = open('data/alumnos.txt','w')

archivo.close()

def leerDara():

print('=====DATOS DE ESTUDIANTE Y 3 NOTAS=====')

nombre = input("Nombre: ")

apellido = input("Apellido: ")

nota1 = int(input("Nota1: "))

nota2 = int(input("Nota2: "))

nota3 = int(input("Nota3: "))

return nombre,apellido,nota1, nota2, nota3

def guardarRegistros(n,a,n1,n2,n3):

archivo=open('data/alumnos.txt','a')

datos=f'{n},{a},{n1},{n2},{n3}\n'

archivo.write(datos)

print('Registro guardado con éxito\n')

print(f'Datos de :{n} {a} en 'alumnos.txt:')

print('_____ \n')

def mostrarRegistro():

if os.path.exists('data/alumnos.txt'):

with open('data/alumnos.txt','r') as archivo:

print('=====MOSRTAR DATOS DE ALUMNOS=====')

for linea in archivo:

print(linea.strip())

def ClasificacionNotas(nota):

if nota >=51:

return 'Aprobado'

else:

return 'Reprobado'

def guardarListaAprobadosRebropbados(n,a,n1,n2,n3):

if n1>=51:

with open('data/aprobados.txt','a') as fileAprobado:

fileAprobado.write(f'{n},{a},{n1}\n')

print(f'Nota1 {n1} de {n} {a} guardada en 'Aprobados.txt:')

else:

with open('data/reprobados.txt','a') as fileRepbrado:

fileRepbrado.write(f'Nota1 {n},{a},{n1}\n')

print(f'Nota1 {n1} de {n} {a} guardada en 'Reprobados.txt:')

if n2>=51:

with open('data/aprobados.txt','a') as fileAprobado:

fileAprobado.write(f'{n},{a},{n2}\n')

print(f'Nota2 {n2} de {n} {a} guardada en 'Aprobados.txt:')

else:

with open('data/reprobados.txt','a') as fileRepbrado:

fileRepbrado.write(f'{n},{a},{n2}\n')

print(f'Nota2 {n2} de {n} {a} guardada en 'Reprobados.txt:')

if n3>=51:

with open('data/aprobados.txt','a') as fileAprobado:

fileAprobado.write(f'{n},{a},{n3}\n')

print(f'Nota3 {n3} de {n} {a} guardada en 'Aprobados.txt:')

A partir del archivo alumnos.txt:

Esteban:Gutierrez:49:18:32

Luisa:Miranda:68:44:99

Jean Paul:Munoz:48:38:81

Gianfranco:Basso:54:54:50

Romina:Smith:100:98:92

crear los siguientes archivos:

aprobados.txt

Luisa, Miranda, 70

Jean Paul, Munoz, 56

Romina, Smith, 97

reprobados.txt

Esteban, Gutierrez, 33

Gianfranco, Basso, 53

else:

with open('data/reprobados.txt','a') as fileRepbrado:

fileRepbrado.write(f'{n},{a},{n3}\n')

print(f'Nota3 {n3} de {n} {a} guardada en 'Reprobados.txt:')

def mostrarListaAprobadoReprobado():

print('=====LISTA DE APROBADOS=====')

if os.path.exists('data/aprobados.txt'):

with open('data/aprobados.txt','r') as archivo:

if archivo:

for linea in archivo:

print(linea.strip())

else:

print('No hay estuidanmtes Aprobados.')

else:

print("No se ha generado el archivo de aprobados.")

print('=====LISTA DE REPROBADOS=====')

if os.path.exists('data/reprobados.txt'):

with open('data/reprobados.txt','r') as archivo:

if archivo:

for linea in archivo:

print(linea.strip())

else:

print('No hay estudiantes Reprobados.')

else:

print("No se ha generado el archivo de reprobados.")

#_____Function Main_____

nombre,apellido,nota1,nota2,nota3 = leerDara()

crearArchivo()

guardarRegistros(nombre,apellido,nota1,nota2,nota3)

mostrarRegistro()

guardarListaAprobadosRebropbados(nombre,apellido,nota1,nota2,nota3)

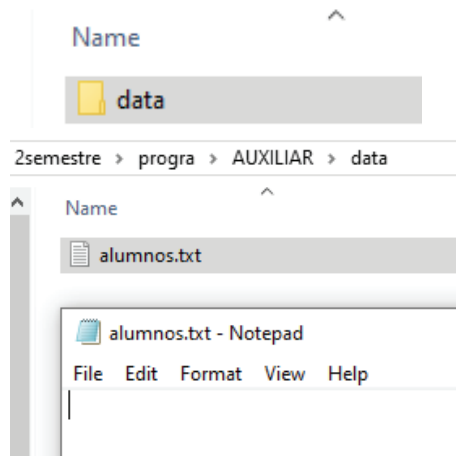
mostrarListaAprobadoReprobado()



DESK CHECK

CREACION DEL ARCHIVO

El archivo "alumnos.txt" no existe en la carpeta "data". Se creará.



=====DATOS DE ESTUDIANTE Y 3 NOTAS=====

Nombre:Romina
Apellido:Smith
Nota1:100
Nota2:98
Nota3:92

El archivo "alumnos.txt"
ya existe en la carpeta "data".
Registro guardado con éxito

Datos de :Romina Smith en 'alumnos.txt'.

=====MOSRTAR DATOS DE ALUMNOS=====

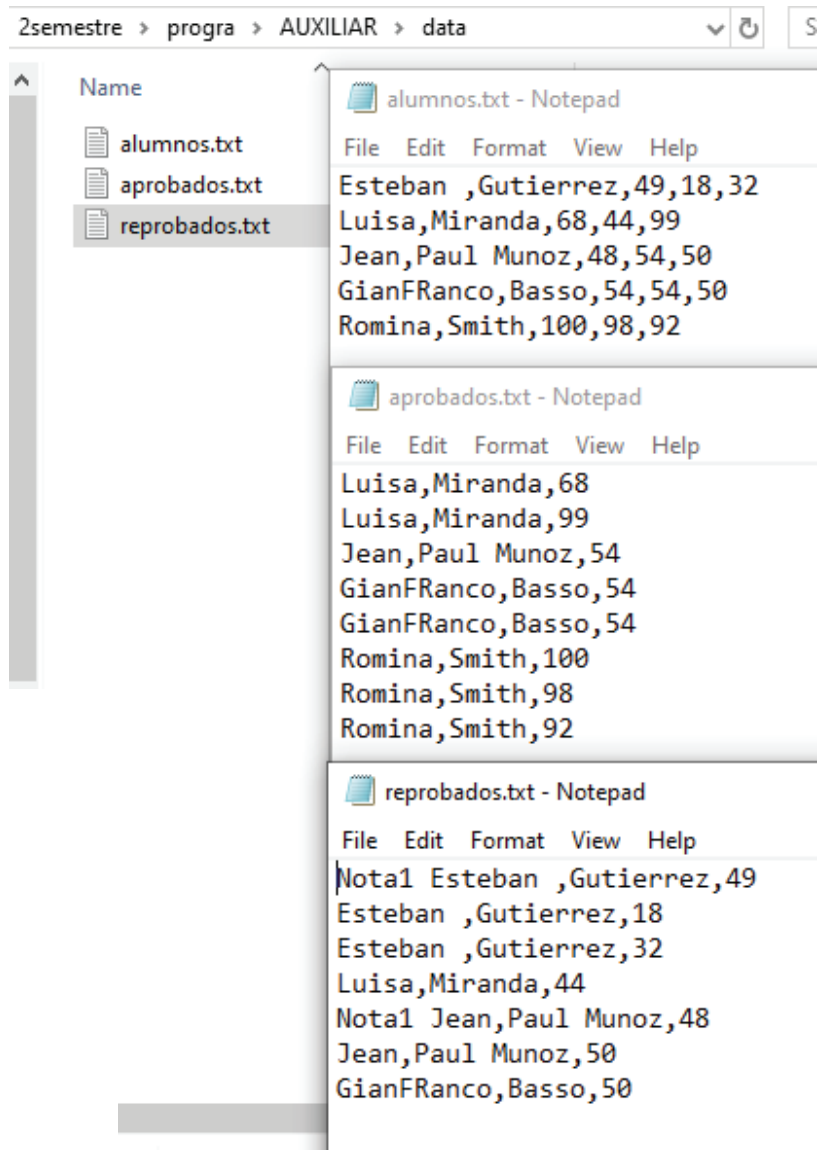
Esteban ,Gutierrez,49,18,32
Luisa,Miranda,68,44,99
Jean,Paul Munoz,48,54,50
GianFRanco,Basso,54,54,50
Romina,Smith,100,98,92
Nota1 100 de Romina Smith guardada en 'Aprobados.txt'.
Nota2 98 de Romina Smith guardada en 'Aprobados.txt'.
Nota3 92 de Romina Smith guardada en 'Aprobados.txt'.

=====LISTA DE APROBADOS=====

Luisa,Miranda,68
,Luisa,Miranda,99
,Jean,Paul Munoz,54
,GianFRanco,Basso,54
,GianFRanco,Basso,54
Romina,Smith,100
,Romina,Smith,98
,Romina,Smith,92

=====LISTA DE REPROBADOS=====

Nota1 Esteban ,Gutierrez,49
,Esteban ,Gutierrez,18
Esteban ,Gutierrez,32
,Luisa,Miranda,44
Nota1 Jean,Paul Munoz,48
Jean,Paul Munoz,50
GianFRanco,Basso,50





PROBLEMA

7. Dado el vector V de tamaño N, con elementos enteros positivos. Ordenar sus elementos ascendentemente, aplicando el Método de Inserción.

SOURCE IN PYTHON

```
class Estudiante:
    def __init__(self, ru, apellidos, nombres, primerParcial, segundoParcial, tercerParcial):
        self.ru = ru
        self.apellidos = apellidos
        self.nombres = nombres
        self.primerParcial = primerParcial
        self.segundoParcial = segundoParcial
        self.tercerParcial = tercerParcial

    def leer(self):
        self.ru = input("Ingrese el registro universitario: ")
        self.apellidos = input("Ingrese los apellidos: ")
        self.nombres = input("Ingrese los nombres: ")
        self.primerParcial = int(input("Ingrese la nota del primer parcial: "))
        self.segundoParcial = int(input("Ingrese la nota del segundo parcial: "))
        self.tercerParcial = int(input("Ingrese la nota del tercer parcial: "))

    def mostrar(self):
        print(f"Registro Universitario: {self.ru}")
        print(f"Apellidos: {self.apellidos}")
        print(f"Nombres: {self.nombres}")
        print(f"Primer Parcial: {self.primerParcial}")
        print(f"Segundo Parcial: {self.segundoParcial}")
        print(f"Tercer Parcial: {self.tercerParcial}")

    def notaTotal(self):
        total = self.primerParcial + self.segundoParcial + self.tercerParcial
        print(f"La suma total de las notas es: {total}")

    def promedio(self):
        promedio = (self.primerParcial + self.segundoParcial + self.tercerParcial) / 3
        print(f"El promedio de las tres notas es: {promedio}")
```

DESK CHECK

```
estudiante = Estudiante("", "", "", 0, 0, 0)
estudiante.leer()
estudiante.mostrar()
estudiante.notaTotal()
estudiante.promedio()
```

```
Ingrese el registro universitario: 200088118
Ingrese los apellidos: Porfiro Elias
Ingrese los nombres: 45
Ingrese la nota del primer parcial: 78
Ingrese la nota del segundo parcial: 78
Ingrese la nota del tercer parcial: 50
Registro Universitario: 200088118
Apellidos: Porfiro Elias
Nombres: 45
Primer Parcial: 78
Segundo Parcial: 78
Tercer Parcial: 50
La suma total de las notas es: 206
El promedio de las tres notas es: 68.66666666666667
```





INGENIERÍA DE SISTEMAS

UNIVERSIDAD PÚBLICA DE EL ALTO

MATERIA : SIS-211 PROGRAMACIÓN I

PRÁCTICA FINAL DE AUXILIADURA

NOMBRE : PORFIRIO ELIAS QUISPE QUISPE

PARALELO : 2 " A "

TURNO : MAÑANA

GESTION : II-2024

DOCENTE : LIC. PEREZ MARTINEZ KATYA MARICELA

AUXILIAR : MIGUEL ANGEL QUISPE GUTIERREZ