

Taller de Programación

Examen

28 de noviembre de 2023

Nombre:.....

Instrucciones:

- Lea atentamente el enunciado de cada uno de los problemas.
- Los problemas 1 y 2 son **obligatorios**. Debe escoger entre el problema 3 y 4.
- Para cada problema cree un archivo.py distinto. El nombre del archivo debe ser el número del problema (uno.py, dos.py, tres.py, cuatro.py).
- Comprima los problemas en un solo archivo ZIP y súbalo a la sección Evaluación en <http://canvas.udd.cl>. Solo tiene una oportunidad para subir sus respuestas.
- No se puede utilizar librerías adicionales ni estilos de sintaxis o métodos avanzados de programación que no se hayan revisado en clases o que no domine.
- En caso de que su profesor(a) tenga dudas sobre la originalidad del código, podrá interrogarlo oralmente para verificar que no hubo faltas a la ética.
- De detectarse copia entre pares o cualquier otra situación que de cuenta de falta a la ética, su certamen será evaluado con la NOTA MÍNIMA y será reportado(a) al comité de ética de la Facultad, lo que puede derivar en una causal de eliminación de la universidad.

Ejercicio 1 Obligatorio: Registro de Ventas de una Librería (2.5 pts)

Desarrolla un programa que gestione el registro de ventas de una librería y genere un resumen específico. El archivo de registro de ventas contiene una fila por libro y tiene la siguiente estructura:

Columnas: Fecha, Título del libro, Autor, Precio.

Cree un programa que lea el archivo CSV `ventas_libreria.csv` y escriba un archivo

llamado `resumen_ventas_Coelho.csv` .

Este archivo debe contener el autor, título del libro, y cantidad total vendida **sólo para el autor "Coelho"**.

Ejemplo de archivo de salida:

```
Autor;Titulo;CantidadTotal
Coelho;Brida;4
Coelho;El Alquimista;12
```

```
with open ('ventas_libreria.csv', 'r') as f: #abrir archivo 0.1 pts
    dicc_count=dict() #crear estructura de edatos para almacena 0.1 pts
    next(f)
    for line in f:
        aux=line.split(';') #separar y seleccionar columnas 0.5 pts
        autor=aux[2]
        libro=aux[1]
        print(aux)
        if autor=='Coelho': #contar correctamente 0.8 pts
            if libro in dicc_count:
                dicc_count[libro]+=1

            else:
                dicc_count[libro]=1

    print(dicc_count)

#escribir correctamente el archivo
with open ('resumen_ventas_Coelho.csv','w') as g:# crear archivo 0.1 pts
    g.write('Autor;'+'Titulo;'+'cantidadTotal' + '\n') #escribir el encabezado
    for libro, numero in dicc_count.items(): #iterar sobre el diccionario 0.3 pts
        g.write('Coelho;' +libro+ ';' + str(numero) + '\n') #escribir correctamente
```

Ejercicio 2 Obligatorio: Dicionarios y Funciones (2.5 pts)

Crea una **función** `contar_palabras_inician_vocal` que tome una frase como entrada y devuelva un diccionario donde las claves sean las palabras presentes en la frase que empiezan con una vocal y los valores sean la cantidad de veces que cada palabra aparece.

Nota: - La función `letra.lower()` en el caso de que una letra sea mayuscula la transforma a una letra minuscula.

```
# Solución Correcta 2.5 pts
def contar_palabras_inician_vocal(texto): #iniciar creacion de funcion y definici
    dicc_cuentas=dict()
    #pasar a mayuscula y/o seleccionar solo digitos 0.2 pts
    texto=texto.lower()
    texto_lista=texto.split()
    print(texto_lista)

    for palabra in texto_lista: #recorrer el texto 0.3 pts
        if palabra[0] in ['a', 'e', 'i', 'o', 'u']:
            if palabra in dicc_cuentas:#llena el diccionario correctamente 1.0
                dicc_cuentas[palabra]+=1
            else:
                dicc_cuentas[palabra]=1

    return dicc_cuentas #retornar diccionario 0.2 pts

texto_prueba='La inteligencia artificial es muy inteligente pero menos inteligente'
print(contar_palabras_inician_vocal(texto_prueba)) #usar correctamente la funci
```

Ejercicio 3: Manejo de Tuplas (1.0 pt)

Escribe un programa que solicite al usuario ingresar N coordenadas en formato `x, y`. Luego, almacena estas coordenadas en una lista de tuplas y luego muestra por pantalla los valores obtenidos.

Ejemplo de entrada: `2,3`

```
# Solución Correcta 1.0 pt
n=4
lista_final=[]
for i in range(n):
    punto=input('Ingrese punto separado por coma:') #solicitar info correctamen
    punto_lista=punto.split(',') #separar correctamente 0.4 pts
    lista_final.append(tuple(punto_lista)) #agregar a lista 0.2 pts

for punto_tupla in lista_final: #desempaquetar tuplas 0.2 pts
    print(punto_tupla)
```

Ejercicio 4: Condicionales y Ciclos (1.0 pt)

Desarrolla una **función** `suma_raiz_cuadrada_pares` que reciba un número `n` y calcule la suma de las raíces cuadradas de todos los números pares desde 1 hasta 'n'.

Recuerde que un número par puede ser representado por la expresión $2k$. Y que el método `sqrt(numero)` de la librería `math` permite calcular la raíz cuadrada de un número.

Solución Parcialmente Correcta 0.8 pts

```
from math import sqrt
def suma_raiz_cuadrada_pares(n):
    suma=0
    for k in range(n+1):
        suma+=sqrt(2*k)
    return suma

n=10
print(suma_raiz_cuadrada_pares(n))
```

Alternativa 2 Solución Correcta 1.0 pts

```
from math import sqrt
def suma_raiz_cuadrada_pares2(n): #definicion correcta 0.1 pts
    suma=0
    for i in range(n+1): #uso de ciclo correcto 0.1 pts
        if i%2==0:
            termino=sqrt(i)
            suma+=termino #suma correcta 0.5 pts
    return suma

n=10
print(suma_raiz_cuadrada_pares2(n)) #uso correcto de la funcion 0.3 pts
```

Alternativa 3 Solución Correcta 1.0 pts

```
from math import sqrt
def suma_raiz_cuadrada_pares3(n): #definicion correcta 0.1 pts
    suma=0
    for k in range(n+1): #uso de ciclo correcto 0.1 pts
        termino=2*k
        if termino<=n+1:
            termino2=sqrt(termino)
            suma+=termino2 #suma correcta 0.5 pts
    return suma

n=10
print(suma_raiz_cuadrada_pares3(n)) #uso correcto de la funcion 0.3 pts
```

////////////////////////////////////