

**FACULTAD DE INGENIERÍA  
FUNDAMENTOS DE COMPUTACION Y PROGRAMACION**

**PRUEBA ACUMULATIVA**



**ASPECTOS GENERALES DE LA PRUEBA**

- Lea atentamente la prueba y las instrucciones antes de comenzar a desarrollarla.
- Queda prohibido hablar con los compañeros(as) durante el desarrollo de la PEP.
- La PEP contiene 2 preguntas de desarrollo, con un total de **45** puntos y una exigencia del **60%**
- Tiene un límite de tiempo de **90** minutos para responder.
- El equipo docente tiene la prohibición de responder consultas.
- El/La estudiante que se sorprenda en actos deshonestos será calificado con la nota mínima.
- Los elementos tecnológicos deben permanecer apagados y guardados. Queda absolutamente prohibido el uso todo elemento tecnológico. Su uso puede significar la nota mínima o sanciones mayores.
- El alumno deberá identificarse con su Cédula de Identidad.
- Sobre el escritorio sólo podrá existir lápiz (obligatorio) y goma/lápiz corrector (opcional).
- Complete sus datos personales antes de comenzar la evaluación.
- Considere que la evaluación contempla el código, los comentarios y el seguimiento de las buenas prácticas de programación.
- Responda cada pregunta, a continuación de su enunciado, en el espacio que se le entrega para ello.

NOMBRE	RUT	SECCIÓN	

**1. (30 puntos) CENSURA**

La CIA ha decidido liberar ciertos documentos previamente clasificados, en su afán constante por “transparentar” sus labores. Estos documentos son antiguos informes realizados por sus operadores, pero antes de desclasificarlos, deben someterlos a un estricto protocolo de censura, para proteger la identidad tanto de sus agentes, como sus asociados. Para realizar este procedimiento, se le pide a usted que desarrolle un programa que realice la censura.

Para este propósito, el programa debe aceptar un nombre de documento, que leerá como archivo, y el nombre de otro archivo que contendrá los nombres y apellidos a censurar por separado, uno por línea, como se muestra en el cuadro a continuación. El programa recorre las palabras, y cuando encuentra un nombre que censurar, reemplaza todos los caracteres del nombre, salvo el primero, por asteriscos (\*). Finalmente, guarda el texto modificado en un nuevo archivo, cuyo nombre es el “nombre original”+Censurado.txt

documento.txt	censurar.txt	documentoCensurado.txt
The ranking Egyptian representative in Washington, a Dr. Ghorbal who sits in the Indian Embassy, professed surprise at the sharp American reaction to the newspaper Al Ahram's recent vilification of US policy. He told Deputy Assistant Secretary Davis	Bruce Ghorbal Davis Mike Strangelove Lovelace Alex Mason Peter Sellers	The ranking Egyptian representative in Washington, a Dr. G***** who sits in the Indian Embassy, professed surprise at the sharp American reaction to the newspaper Al Ahram's recent vilification of US policy. He told Deputy Assistant Secretary D****

that such publishing practices were not something unknown in other countries, even in the US. Ghorbal claimed, however, that Egypt had acted to end this chapter before a new administration took over. He hoped the two countries could now look forward, not backward to past differences.	Kong Turgidson	that such publishing practices were not something unknown in other countries, even in the US. G***** claimed, however, that Egypt had acted to end this chapter before a new administration took over. He hoped the two countries could now look forward, not backward to past differences.
--	----------------	---

Note que los textos contienen puntos y comas, por lo que debe tenerlo en consideración.

**Restricción:** No se puede utilizar el método `.replace(str, str)`.

```
# Función que lee un archivo de texto
# Entrada: nombre de archivo (string)
# Salida: contenido del archivo (string)
def leerArchivo(nombreArchivo):
    # Se abre el archivo en modo de lectura
    archivo = open(nombreArchivo, 'r')
    # Se crea un string vacío para almacenar el resultado
    contenido = ''
    # Para cada línea en el archivo
    for linea in archivo :
        # Se agrega esta al string
        contenido += linea
    # Se cierra el archivo
    archivo.close()
    # Se retorna el contenido del archivo
    return contenido

# Función que escribe el contenido en un archivo de texto
# Entrada: Nombre de archivo (string), contenido a escribir (string)
# Salida: True, si la escritura fue exitosa
def escribirArchivo(nombreArchivo, contenido):
    # Se separa el nombre del archivo por el caracter punto
    nombreNuevo = nombreArchivo.split('.')
    # Se genera el string con el nombre del nuevo archivo
    nombreArchivo = nombreNuevo[0] + 'Censurado.' + nombreNuevo[1]
    # Se abre el archivo en modo de escritura
    archivo = open(nombreArchivo, 'w')
    # Se escribe el contenido del archivo
    archivo.write(contenido)
    # Se cierra el archivo
    archivo.close()
    # Se retorna True para indicar que el proceso finalizó sin errores
```

```

return True

# Función que revisa si la palabra debe ser censurada y la censura en caso de
# ser necesario
# Entrada: Palabra a revisar (string), palabras a censurar (lista)
# Salida: Palabra con censura (string)
def aplicarCensura(palabra, palabras):
    # Se crea un string vacío para almacenar los signos de puntuación en caso
    # de existir
    punto = ''
    # Si la palabra posee un signo de puntuación
    if palabra[-1] == '.' or palabra[-1] == ',':
        # Se almacena el signo
        punto = palabra[-1]
        # Se eliminan los signos del final de la palabra usando strip
        palabra = palabra.strip('.').strip(',')
    # Si la palabra está en la lista de palabras a censurar
    if palabra in palabras:
        # Se aplica la censura
        palabra = palabra[0] + '*' * (len(palabra) - 1)
    # Se retorna la palabra con el signo de puntuación original
    return palabra + punto

# Función que realiza la censura
# Entrada: Contenido a censurar (string), palabras a censurar (string)
# Salida: Contenido censurado (string)
def censurarContenido(contenido, palabras):
    # Se separa el archivo a censurar en párrafos con split
    contenido = contenido.split('\n')
    # Se separa el archivo de palabras a censurar con split
    palabras = palabras.split('\n')
    # Se declara un iterador para ir separando los párrafos en palabras
    i = 0
    # Mientras i no alcance el largo del contenido,
    # en este caso, el número de párrafos
    while i < len(contenido):
        # Se convierte cada párrafo (string) en una lista de
        # palabras
        contenido[i] = contenido[i].split(' ')
        # Se incrementa el iterador
        i = i + 1
    # Para almacenar el resultado de la censura, se crea un string
    # vacío para almacenar el texto
    resultado = ''
    # Para cada párrafo del texto
    for parrafo in contenido:
        # Para cada palabra en el párrafo
        for palabra in parrafo:
            # Se aplica la censura a la palabra

```

```

        palabra = aplicarCensura (palabra, palabras)
        # Se añade esta al texto
        resultado = resultado + ' ' + palabra
        # Se agrega el párrafo y su respectivo salto de línea
        resultado = resultado.strip(' ') + '\n'
    # Se retorna el contenido censurado una vez finalizado el proceso
    return resultado

# BLOQUE PRINCIPAL

# ENTRADA

# Se solicitan los nombres del archivo a censurar y del archivo con la lista
# de palabras a censurar
archivoACensurar = raw_input('Ingrese nombre de archivo a censurar: ')
archivoPalabras = raw_input('Ingrese el nombre del archivo con palabras: ')

# Se lee el contenido de ambos archivos
contenidoACensurar = leerArchivo(archivoACensurar)
palabrasACensurar = leerArchivo(archivoPalabras)

# PROCESAMIENTO

# Se censura el contenido del archivo
contenidoCensurado = censurarContenido(contenidoACensurar, palabrasACensurar)

# SALIDA

# Se escribe el resultado en el archivo, y si el proceso arroja un True
if escribirArchivo(archivoACensurar, contenidoCensurado) :
    # Se informa al usuario que la escritura fue realizada exitosamente
    print 'El archivo fue escrito exitosamente'

```

## 2. (15 puntos) Mínimos cuadrados

El método de los mínimos cuadrados dice que si se tienen  $n$  puntos en pares ordenados  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ , estos pueden ajustarse a una recta de tipo:

$$y = a + bx$$

a través de la fórmula

$$v \approx (M^T M)^{-1} M^T y$$

dónde

$$v = \begin{pmatrix} a \\ b \end{pmatrix}, \quad M = \begin{pmatrix} 1 & x_0 \\ 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \quad y = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

El vector  $v$  resultante de la operación, permite aproximar los valores estimativos  $a$  y  $b$  de la recta.

Se pide que, mediante un programa en Python, encuentre la curva que mejor ajusta los datos de la medición que se presenta en la tabla y que genere un gráfico discreto, con dichos puntos como asteriscos que no deben unirse y la recta que ajusta la medición con una línea roja.

$x_n$	$y_n$
6.1	0
7.6	2
8.7	4
10.4	6

Consideraciones:

- Considere para su trabajo que una variable matrix de tipo array se puede transponer mediante `matrix.T` (ej: `trans = matrix.T`).
- El módulo Numpy tiene una función que puede llamarse mediante `numpy.linalg.inv(x)` que calcula la matriz inversa de  $x$ .

```
# IMPORTACIÓN DE MÓDULOS
# Importamos el módulo numpy con alias np
import numpy as np
# Importamos el módulo matplotlib.pyplot con alias plt
import matplotlib.pyplot as plt

## ENTRADA
# Definimos las listas de entrada
# Las cuáles corresponden a valores estáticos
xDeN = [6.1, 7.2, 8.7, 10.4]
yDeN = [0, 2, 4, 6]

## PROCESAMIENTO
```

```

# Definimos la matriz M, que consiste de filas con compones 1 y xDeN
M = []
for num in xDeN:
    M.append([1,num])

# Creamos matrices de numpy
x = np.array(xDeN)
y = np.array(yDeN)
M = np.array(M)

# Calculamos el producto entre la matriz transpuesta y la matriz original
prod = (M.T.dot(M))
# Calculamos la inversa del producto por la matriz transpuesta
vIntermedio = np.linalg.inv(prod).dot(M.T)
# Finalmente, el vector v es el producto anterior por el vector y
v = vIntermedio.dot(y)

# Creamos un vector x de 50 puntos con rango que sale un poco de la tabla
# original
vectorX = np.linspace(x[0]-1, x[-1]+1)
# Creamos el vector y a través de la fórmula de la recta.
vectorY = v[0] + vectorX * v[1]

## SALIDA

# Gráfico de puntos
puntos = plt.plot(x, y)
# La línea
recta = plt.plot(vectorX, vectorY)

# Eliminamos la línea y colocamos puntos grandes
plt.setp(puntos, "linestyle", "none", "marker", "*")
# Aseguramos que el segundo gráfico sea una línea
plt.setp(recta, "linestyle", "-")

# Configuramos título y ejes
plt.title("Ajuste de recta")
plt.xlabel("x")
plt.ylabel("y")
# Mostramos
plt.show()

```