

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN  
CCPG1001 - FUNDAMENTOS DE PROGRAMACIÓN  
TERCERA EVALUACIÓN - I TÉRMINO 2016-2017/ Septiembre 13, 2016

Nombre: \_\_\_\_\_ Matrícula: \_\_\_\_\_  
Paralelo: \_\_\_\_\_

COMPROMISO DE HONOR: Al firmar este compromiso, reconozco que el presente examen está diseñado para ser resuelto de manera individual, que puedo usar un lápiz o esferográfico; que sólo puedo comunicarme con la persona responsable de la recepción del examen; y, cualquier instrumento de comunicación que hubiere traído, debo apagarlo y depositarlo en la parte anterior del aula, junto con algún otro material que se encuentre acompañándolo. Además, no debo usar calculadora alguna, consultar libros, notas, ni apuntes adicionales a los que se entreguen en esta evaluación. Los temas debo desarrollarlos de manera ordenada. Firmo el presente compromiso, como constancia de haber leído y aceptado la declaración anterior. "Como estudiante de ESPOL me comprometo a combatir la mediocridad y actuar con honestidad, por eso no copio ni dejo copiar".

\_\_\_\_\_  
Firma

**TEMA 1 (30 PUNTOS)**

Una empresa de telecomunicaciones necesita automatizar el cálculo del costo de enviar un mensaje basado en el número de palabras, el tamaño de cada palabra y el tipo de palabra. Para el cálculo:

1. se determina que una palabra corta tiene máximo M caracteres
2. se determina que una palabra larga tiene más de M caracteres
3. se define como palabra especial los verbos en infinitivo, es decir, palabras terminadas en "ar", "er", "ir", sin importar su tamaño.

Para calcular el costo total del mensaje usted debe cobrar un valor por las palabras cortas, otro valor por las palabras largas y otro valor por los infinitivos. Suponga que todos los costos para este problema están dados en dólares americanos.

Implemente las siguientes funciones:

- a. **cargarDatos(nombreArchivo)** que recibe el nombre del archivo que especifica en líneas separadas el tamaño M, el costo de las palabras cortas, el costo de las palabras largas, y el costo de los infinitivos. Su función deberá leer este archivo y retornar un diccionario con el siguiente formato del ejemplo:

```
10
0.2
0.5
0.3
```

Costos.txt

cargarDatos('Costos.txt') retorna

{'M':10, 'Corta':0.2, 'Larga':0.5, 'Infinitivo':0.3}

- b. **calcularCostos(datos, nombreArchivo)** que recibe el diccionario de datos generado en el literal a) y un nombre de archivo. El archivo contiene el texto del mensaje grabado línea por línea. Además, cada línea contiene múltiples palabras separadas por espacios. El único signo de puntuación presente en el texto es un punto '.' al final del mensaje y no deberá ser considerado para determinar el costo de esa palabra. La función debe retornar el costo total del mensaje.
- c. **cambiarMensaje(datos, nombreArchivo1, nombreArchivo2)** que recibe el diccionario de datos generado en el literal a) y dos nombres de archivos. La función debe leer el texto del mensaje de *nombreArchivo1* y escribir en *nombreArchivo2* el nuevo mensaje, acortando las palabras largas a M - 1 caracteres y colocando '#' al final de cada una de ellas, y reemplazando el punto final con la palabra especial 'END'.

## TEMA 2 (60 PUNTOS)

Una empresa agrícola ha decidido integrar un dron (avión no tripulado) a una de sus plantaciones de área  $M \times N$  para poder monitorear el crecimiento de sus cultivos. El dron a utilizarse tiene la capacidad de sensar el número de cultivos en una posición  $(i,j)$  por medio de la función ***sensarCultivos(tuplaPosicion)*** que mueve el dron a la posición dada por la tupla y retorna un valor entero correspondiente al número de cultivos en dicha posición. Suponga que esta función ya existe, por lo tanto, no necesita ser desarrollada por usted.

Implemente las siguientes funciones:

- a. ***generarPlantacion (tuplaDimensionPlantacion)*** que recibe una tupla indicando el tamaño total  $(M,N)$  de la plantación y procede a sensar los cultivos utilizando ***sensarCultivos***. La función retorna la matriz **plantación** que indica el número de cultivos en cada posición de la plantación.
- b. ***analizarDensidad(plantación, limite)*** que recibe como parámetro la matriz **plantación** del literal a) y un valor entero positivo. La función debe retornar una nueva matriz indicando los grados de crecimiento de la plantación. Una posición  $(i,j)$  de la plantación es considerada con crecimiento 'BAJO' si tiene menos de *limite* cultivos, caso contrario es considerada de crecimiento 'ALTO'. Al definir la función, considere que el valor por defecto de *limite* debe ser 4. Por ejemplo:

Si plantacion =  $\begin{bmatrix} 5 & 3 & 2 \\ 1 & 4 & 8 \\ 2 & 3 & 1 \end{bmatrix}$

**analizarDensidad(plantacion)**, retorna:

$\begin{bmatrix} \text{'ALTO'} & \text{'BAJO'} & \text{'BAJO'} \\ \text{'BAJO'} & \text{'ALTO'} & \text{'BAJO'} \\ \text{'BAJO'} & \text{'BAJO'} & \text{'BAJO'} \end{bmatrix}$

- c. ***reporteCrecimiento(plantacion, densidad)*** que recibe como parámetros las matrices de los literales a) y b). Suponga que "surco" es equivalente a una fila de la matriz y "parcela" es equivalente a una celda de la matriz. La función debe retornar una tupla de tres elementos con la siguiente información:

1. Los promedios de cultivos por surcos.
2. Las posiciones, relativas a cada surco, de las parcelas que tienen el mayor número de cultivos en dicho surco.
3. Los promedios de cultivos de las parcelas para los grados de crecimiento 'ALTO' y 'BAJO'

Por ejemplo, si plantacion =  $\begin{bmatrix} 5 & 3 & 2 \\ 1 & 4 & 8 \\ 2 & 3 & 1 \end{bmatrix}$

**reporteCrecimiento(plantacion, densidad)**, retorna:

( [3.33333333, 4.33333333, 2.0], [0, 2, 1], [5.66667, 2.0] )

### TEMA 3 (10 PUNTOS)

- a. Considere el siguiente código e indique qué se muestra por pantalla. Justifique su respuesta (5 puntos)

```
lista1 =[3, 'A', 6]
lista2 =['A']

def funcion (lista1, lista2):
    a=[]
    for i in lista1:
        for j in lista2:
            if i != j:
                a.append(str(i) + str(j))
                for x in a[:]:
                    a.append(str(i) + str(j))
    return a

print(funcion (lista1, lista2))
```

- b. Considere el siguiente código e indique qué se muestra por pantalla. Justifique su respuesta (5 puntos)

```
def fun(cadena, k):
    L = []
    for elem in set(cadena.split(' ')):
        L.append(elem * k)

    return '#'.join(L)

cadena = 'programar es estupendo estupendo es programar'
print(fun(cadena, 2))
```