

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
CCPG1001 - FUNDAMENTOS DE PROGRAMACIÓN
TERCERA EVALUACIÓN - I TÉRMINO 2019-2020/ Septiembre 13, 2019

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 (60 PUNTOS)

Asuma que tiene un archivo con la información de los productos agrícolas cosechados por una hacienda durante todos los días del año 2018. La información se encuentra en el siguiente formato:

```
codigo1,codigo2,codigo3,...,codigoN  
codigo_producto,fecha(dd-MMM-aaaa),cantidad_cosechada
```

Ejemplo:

```
100034,100312,100021,...,201245,432198  
codigo_producto,fecha(dd-MMM-aaaa),cantidad_cosechada  
100034,02-ENE-2018,5  
100021,02-ENE-2018,15  
100021,07-ENE-2018,11  
432198,20-ENE-2018,12  
...
```

Nota: La primera línea del archivo contiene los códigos de todos los productos agrícolas presentes en el resto del archivo, mientras que la segunda línea es la cabecera del archivo.

1. [8 puntos] **crearMatriz(nomArchivo)** que recibe el nombre del archivo con la información de las cosechas de un año; y devuelve un vector con todos los códigos de productos y una matriz con los totales (valores enteros) de cosechas para cada producto (filas) durante cada mes del año (columnas). Ejemplo:

Cod	Y											
	ENE	FEB	MAR	ABR	MA	JUN	JUL	AGO	SEP	OCT	NOV	DIC
100034	32	12	45	67	84	114	21	57	99	84	74	65
100312	43	15	67	21	77	95	110	78	93	63	56	32
100021	65	78	32	155	32	73	87	91	22	65	82	17
...												
201245	39	71	63	32	57	85	83	12	11	15	34	65
432198	55	51	54	67	64	63	56	52	71	77	87	32

2. **[5 puntos] mesMenosRentable(M)** que recibe la matriz de cosechas **M**. Esta función retorna el **nombre del mes** en que menos se cosechó y el **total de cosecha** de ese mes.
3. **[8 puntos] altoBajos(M, k)** que recibe la matriz de cosechas **M** y un entero **k**. La función retorna el nombre de todos los meses que tienen una cosecha total con al menos **k** unidades por arriba de la cosecha del peor mes del año.
4. **[9 puntos] peorTrimestre(M, Cod, codigo)** que recibe la matriz de cosechas **M**, el vector de códigos **Cod** y el **codigo** de un producto. La función debe retornar el nombre del trimestre ("T1", "T2", "T3" o "T4") en el que menos se cosechó el producto con **codigo**.
5. **[6 puntos] peoresNProductos(M, Cod, n)** que recibe la matriz de cosechas **M**, el vector de códigos **Cod** y un número entero **n**. La función debe retornar los códigos de los **n** productos menos cosechados durante el año.
6. **[9 puntos] promedioProductos(M, Cod, codigos)** que recibe la matriz de cosechas **M**, el vector de códigos **Cod** y una lista con **códigos** de productos. La función retorna el promedio de los totales de cosecha entre los códigos dados en la lista.
7. **[15 puntos] porCategoria(M, Cod, categorias)** que recibe la matriz de cosechas **M**, el vector de códigos **Cod** y un diccionario **categorias** con el siguiente formato:

```
categorias = {'legumbres':[100034,201245,...],
             'verduras':[100021,200013,...],
             ... }
```

La función deberá escribir un archivo por cada categoría con el siguiente formato:

Archivo **legumbres.txt**

```
codigo,ENE,FEB,MAR,ABR,MAY,JUN,JUL,AGO,SEP,OCT,NOV,DIC
100034,32,12,45,67,84,114,21,57,99,84,74,65
201245,39,71,63,32,57,85,83,12,11,15,34,65
...
```

El nombre del archivo debe ser el mismo de la categoría dentro del diccionario **categorias** y terminar con “.txt”.

TEMA 2 (30 PUNTOS)

Escriba un programa de Python que implemente un juego de adivinanzas de palabras.

Considere que para su programa ya están definidas las siguientes variables y función:

1. Una lista **C** de las categorías para el juego.

```
C = ['Transportes', 'Alimentos', 'Deportes', ...],
```

2. La función **perteneceCategoria(palabra, categoria)** que retorna **True** si palabra pertenece a la categoría, **False** en caso contrario.
3. El diccionario **puntajes** con el siguiente formato:

```
puntajes = {'Transportes':{'a':10, 't':4, 'f':5, ...},
            'Deportes':   {'a':3, 'z':5, 't':10, ...},
            ... }
```

Para la implementación del juego considere las siguientes reglas:

1. El jugador tiene cinco turnos para jugar

Para cada turno:

2. El programa selecciona aleatoriamente una categoría de la lista **C**
3. El programa le pide al jugador que ingrese una palabra para la categoría seleccionada en el paso 2
4. Si la palabra pertenece a la categoría dada y no ha sido ingresada en un turno anterior, calcule los puntos totales para la palabra. El puntaje de la palabra es la suma de los puntajes de cada una de sus letras. De acuerdo al diccionario **puntajes**, una letra tendrá puntajes distintos dependiendo de la categoría del paso 2.
 - a. Si la palabra no cumple con las condiciones, el jugador no obtiene puntos en ese turno
5. En cada turno, muestre el puntaje obtenido para la palabra ingresada y el puntaje acumulado

Al final de los cinco turnos:

6. El jugador habrá ganado si completa un mínimo de 910 puntos, muestre el mensaje correspondiente "Ganó" o "Perdió".

TEMA 3 (10 PUNTOS)

¿Qué imprime el siguiente código? Justifique su respuesta

```
A = 'Num empresas,17,0|9|1,10|19|2,20|29|3'
campos = A.split(',')
valores = []
for rango in campos[2:]:
    valores.append(int(rango.split('|')[0]))
print(valores)
```

Asuma que este tema NO tiene errores de compilación. Si usted cree que hay algún error de compilación, consúltelo inmediatamente con su profesor.

---//---

Cheat Sheet. Funciones y propiedades de referencia en Python.

Librería Numpy para arreglos :	para listas :	para cadena s:	Random as rnd :
np.array([elementos],dtype=) np.unique(arreglo) np.sum(arreglo) np.mean(arreglo) np.argmax(arreglo) np.argsort(arreglo) arreglo .shape arreglo .size arreglo .sum()	listas .append(...) listas .extend(...) listas .count(...) listas .index(...) listas .pop() elemento in listas	cadena s.islower() cadena s.isupper() cadena s.lower() cadena s.upper() cadena s.split(...) cadena s.find(...) cadena s.count(...) cadena s.replace(a,b)	rnd .randint() rnd .choice(lista) rnd .sample(lista,cant) rnd .shuffle(lista)