

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
 CCPG1001 - FUNDAMENTOS DE PROGRAMACIÓN
 TERCERA EVALUACIÓN - I TÉRMINO 2017-2018/ Septiembre 16, 2017

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 ESPO me comprometo a combatir la mediocridad y actuar con honestidad, por eso no copio ni dejo copiar".

 Firma

TEMA 1 (45 PUNTOS)

Considere que tiene **varios** archivos de notas cuyos nombres tienen el siguiente formato *notas-año-término.csv* y su contenido es: *matricula,materia,nota_parcial,nota_final,nota_mejoramiento,AP/RP*

Ejemplo:

<i>notas-2015-I.csv</i>	<i>notas-2013-II.csv</i>	...
201521342,Fundamentos de Programación,80,95,99,AP 201321454,Economía,12,21,33,RP 201321454,Fundamentos de Programación,45,76,89,AP ...	200901456,Física I,80,91,73,AP 201321454,Química,45,57,73,AP 201121372,Economía,73,82,83,AP 201321454,Fundamentos de Programación,11,9,9,RP ...	

Realice lo siguiente:

1. **[10 puntos + 5 puntos de bono]** Implemente la función **academico(archivos)** que recibe una tupla con los nombres de los archivos de notas de los estudiantes. La función deberá retornar un diccionario con la siguiente estructura:

```

notas = {201321454: {'2013-II': [('Química', 45, 57, 73, 'AP'),
                                ('Fundamentos de Programación', 11, 9, 9, 'RP'), ...],
                    '2015-I': [('Economía', 12, 21, 33, 'RP'),
                                ('Fundamentos de Programación', 45, 76, 89, 'AP'), ...],
                    ... },
          200901456: {'2013-II': [('Física I', 80, 91, 73, 'AP'), ...],
                    ... },
          ... }
    
```

2. **[5 puntos]** Implemente una función **semestres(notas, matricula)** que recibe el diccionario de notas y un número de matrícula. La función devolverá una **tupla con todos los semestres en que el estudiante ha tomado materias**.
3. **[15 puntos]** Implemente una función **nota_academico(notas, matricula, materia)** que recibe el diccionario de notas, una matrícula y el nombre de una materia. La función devolverá la **nota total** para la materia, que se calcula como el **promedio de las dos notas más altas entre parcial, final y mejoramiento**. Si el estudiante no ha tomado la materia aún, la función deberá retornar cero (0). Si el estudiante ha tomado la materia en más de una ocasión, la función deberá retornar el promedio más alto.
4. **[15 puntos]** Implemente una función **mas_reprobados(notas, semestre)** que devuelve el nombre de la materia con mayor cantidad de estudiantes reprobados para dicho semestre.

TEMA 2 (45 PUNTOS)

Como meteorólogo, Ud tiene registros de todos los huracanes por año del Atlántico Norte. Cada huracán tiene un nombre. Los registros se encuentran en una matriz **M** de NumPy, con la siguiente estructura:

	2016			2017				
	...	Alex	...	Otto	Arlene	Harvey	Irma	
Velocidad desplazamiento (Vd)		7		7	19		15	18
Velocidad Rafagas (Vr)		89		195	120		150	240
Velocidad Viento (vv)	...	65	...	165	100	...	110	200
LLuvia		30		49	35		89	67
Marejada		0.5		1.8	1		2	5

Categorías de huracanes:

1. Vr < 100 kph
 2. Vr de 100 a 150 kph
 3. Vr de 150 a 200 kph
 4. Vr de 200 a 250 kph
 5. Vr > 250 kph
- Las velocidades están en **kilómetros por hora (kph)**
 - La cantidad de lluvia está en **centímetros (cm)**
 - Las alturas de marejadas están en **metros (mts)**

Además tiene un diccionario donde la clave es el año (int) y el valor es una tupla con los nombres de los huracanes para ese año. Los nombres de los huracanes no se repiten y están en el mismo orden en el que aparecen en **M**:

```
huracanes = { 2015: ('Ana', ..., 'Kate'),
              2017: ('Arlene', ..., 'Harvey', 'Irma'),
              2016: ('Alex', ..., 'Otto'),
              ... }
```

Entonces:

1. **[11 puntos]** Implemente la función **total_lluvia(M, cat)** que retorna el total de lluvia en centímetros causadas por los huracanes que tengan categoría **cat**. La categoría es recibida como un valor entre 1 y 5.
2. **[5 puntos]** Implemente la función **indices_año(huracanes, año)** que retorna una tupla con los índices de columna donde empieza y termina **año** en la matriz **M**.
3. **[11 puntos]** Escriba la función **velocidad_superior(M, huracanes, año)** que retorna la cantidad de huracanes en **año** que tienen la Velocidad de Viento (VV) superior a la Velocidad de Viento promedio del **año** dado como parámetro.
4. **[11 puntos]** Implemente la función **ACE(M, huracanes, año)** que devuelve la cantidad de energía liberada por todos los huracanes de la temporada **año**. Esto se calcula así:

$$\text{cantidad_energia} = 10^{-4} + (VV_1^2 * VV_2^2 * \dots * VV_n^2)$$
 Donde VV_i es la Velocidad de Viento (en kph) de cada huracan.
5. **[7 puntos]** Implemente la función **lluvia(M, huracanes, nombre_huracan, año)** que devuelve la cantidad de lluvia en centímetros generada por el **nombre_huracan** en **año**.

TEMA 3 (10 PUNTOS)

a. [5 puntos] ¿Qué imprime el siguiente código? Justifique su respuesta.

```
a = 'mensaje'
c = 'abcdefg'
i = len(c) - 1
for b in a:
    print(b * c.index(c[i]))
    i -= 1
```

b. [5 puntos] ¿Qué imprime el siguiente código? Justifique su respuesta.

```
L = [5, 3, 8, 2, 7]
A = [0] * 10
indice = 3
for valor in L:
    if valor < 6:
        indice -= 1
    else:
        indice += 1
    A[indice] = valor
print(set(A))
```

---//---

Cheat Sheet. Funciones y propiedades de referencia en Python.

Librería Numpy para arreglos :	para listas :	para cadena s:
<code>np.array((numRows,numCols),dtype=)</code> <code>np.empty((numRows,numCols),dtype=)</code> arreglos .shape arreglos .reshape() numpy.sum(arreglos) numpy.mean(arreglos) arreglos .sum(axis=1) arreglos .fill(valor)	listas .append(...) listas .count(...) listas .index(...) listas .pop() listas .sort() <i>elemento</i> in listas	cadena s.islower() cadena s.isupper() cadena s.lower() cadena s.upper() cadena s.split(...) cadena s.find(...) cadena s.count(...)

1. Escriba la función **categoria(M, huracanes, año)**, que devuelve un diccionario con la categoría de cada huracán de un año determinado. Las categorías se basan en la **Velocidad Viento** (revisar valor en **M**):
 - a. Cat 1: < 100 kph
 - b. Cat 2: 100 a 150 kph
 - c. Cat 3: 150 a 200 kph
 - d. Cat 4: 200 a 250 kph
 - e. Cat 5: > 250 kph

La función debe devolver un diccionario con los nombres de los huracanes y las categorías:

[('Irma', 5), ('Harvey', 3), , ('Arlene', 1)]

2. Implemente una función **huracanesMayores(diccionario, cat)**, que recibe un diccionario con el formato del retorno de la función anterior y devuelve el **número** de huracanes con categoría **mayor o igual a cat**.