

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN CCPG1001 - FUNDAMENTOS DE PROGRAMACIÓN PRIMERA EVALUACIÓN - I TÉRMINO 2017-2018/ Julio 1, 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 ESPOL me comprometo a combatir la mediocridad y actuar con honestidad, por eso no copio ni dejo copiar".

Firma

TEMA 1. (40 PUNTOS)

El juego de mesa Scrabble es un juego donde se forman palabras en un tablero, a las cuales se les asigna un puntaje. Las palabras pueden crearse cruzándolas con palabras ya existentes en el tablero. El ganador es quien más puntos haya obtenido:

A su equipo se le ha encargado la implementación de este juego. En particular, tiene que implementar un programa **que asigne puntaje a las palabras**. Cada letra tiene una puntuación, mostrada abajo:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	3	3	2	1	4	2	4	1	9	5	1	3	1	1	3	10	1	1	1	1	4	4	9	4	10

Las letras compartidas entre palabras reciben **el doble de puntos**. Su programa recibirá una **secuencia de palabras, separadas por comas**, ingresada por el usuario por teclado y determinará el puntaje de cada una. Para denotar una letra compartida, la **misma será sucedida por el símbolo "*"**. Asuma que **todas las palabras terminan con una letra compartida**. Todas las letras ingresadas deben ser mayúsculas. **Si se ingresa un letra minúscula, esta es ignorada (puntuación de 0 para dicha letra)**. Finalmente, se debe determinar la palabra con mayor puntaje.

Una corrida ejemplo del programa sería:

Ingrese las palabras a calificar: CAS*A*,S*ASTR*E*,R*EY*,A*ZOTE*

Las calificaciones de las palabras son:

casa: 8

sastre: 9

rey: 11

azote: 16

La palabra con mayor puntaje es AZOTE (16 puntos).

Detalle de cómo se obtuvieron los puntajes de cada palabra:

String ingresado: CAS*A*,S*ASTR*E*,R*EY*,A*ZOTE*

casa: 8 -> 3 + 1 + (2 * 1) + (2 * 1)

sastre: 9 -> (2 * 1) + 1 + 1 + 1 + (2 * 1) + (2 * 1)

rey: 11 -> (2 * 1) + 1 + (2 * 4)

azote: 16 -> (2 * 1) + 10 + 1 + 1 + (2 * 1)

TEMA 2. (50 PUNTOS)

SocialBlade (<https://socialblade.com/youtube/>) nos ha proporcionado los datos para los youtubers de tres diferentes países (España, Ecuador y México). Asuma que tiene las siguientes tres listas con los nombres de usuario de los youtubers de cada país.

españa = ['elrubiusOMG', 'VEGETTA777' ... "Pablo Alborán"]

ecuador = ['enchufetvLIVE', 'Kreizivoy', ... "Ecuavisa"]

mexico = ['Yuya', 'Werevertumorro', ... "CaELiKe"]

Adicionalmente se tiene una matriz **M** con datos específicos de *suscriptores*, *reproducciones de vídeos*, *ganancias mensuales* y *ganancias anuales*, en ese orden, para cada youtuber:

	España			Ecuador			México		
	elrubiusOMG	VEGETTA777	..	enchufetvLIVE	Kreizivoy	..	Yuya	Werevertumorro	..
Suscriptores	24771906	18451280	..	78493	133538	..	18554394	13548964	..
Reproducciones de vídeos	5477807839	7046108694	..	798122	21104851	..	1967543913	2034702069	..
Ganancias mensual	21900	45500	..	36	156	..	6700	12200	..
Ganancias anuales	262800	546000	..	430	1900	..	80000	12200	..

Considere las siguientes métricas:

- Popularidad = número de suscriptores
- Rentabilidad = ganancias anuales / número de suscriptores

La compañía Líderes le ha pedido un programa que calcule lo siguiente:

(Para los numerales 1 al 5, su código no deberá mostrar nada por pantalla, solo generar los valores con el tipo de dato pedido y almacenarlos en variables)

1. Nombres de usuarios de los youtubers con mayor *rentabilidad* en cada país. *Tipo de dato de respuesta: lista de strings*.
2. El nombre del país del youtuber con la mayor *rentabilidad*. *Tipo de dato de la respuesta: string*
3. Cuántos youtubers de España tienen más suscriptores que el youtuber más *popular* de América (Ecuador y México). *Tipo de dato de respuesta: valor entero*.
4. El número promedio de reproducciones de los youtubers con más de 1'000,000 de suscriptores. *Tipo de dato de respuesta: valor entero*.
5. Cuántos youtubers de Ecuador hay en cada categoría. La categoría se calcula en base a la siguiente tabla: *Tipo de dato de respuesta: ndarray de enteros*.

Rango de rentabilidad	Categoría
0.0 a 0.30	3
0.31 a 0.60	2
> 0.61	1

6. El *país* que generó más ganancias anuales y el *país* que generó menos ganancias anuales. Luego muestre el siguiente mensaje reemplazando los datos apropiadamente:

El país X generó Z% más de ganancias que el país Y.

Para calcular el porcentaje utilice la siguiente fórmula.

GX: ganancias anuales del país X

GY: ganancias anuales del país Y (GY es menor que GX)

Porcentaje = $(GX - GY) / GY * 100$

TEMA 3. (10 PUNTOS)

a) Dada la siguiente lista `L = [12, 9 , 1, 3, 2, 10, 20, 5, ...]`

Genere 3 posiciones aleatorias de tal manera que los valores correspondientes sumen al menos 20. Al final imprima la suma de los 3 valores seleccionados aleatoriamente.

b) Dada la siguiente lista:

```
stars=['Potter', 'Ron Weasley', 'Dumbledore', 'Hermione Granger','Hagrid','Voldemort']
```

¿Qué imprime el siguiente código?

i.) `print(stars[-4:-2])`

ii.) `print(stars[3][0:stars[3].find(" ")] + stars[1][3:])`

---//---

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>arreglos.shape</code> <code>arreglos.reshape()</code> <code>numpy.sum(arreglos)</code> <code>numpy.mean(arreglos)</code> <code>arreglos.sum(axis=1)</code>	<code>listas.append(...)</code> <code>listas.count(...)</code> <code>listas.index(...)</code> <code>listas.pop()</code> <code>elemento in listas</code>	<code>cadena.islower()</code> <code>cadena.isupper()</code> <code>cadena.lower()</code> <code>cadena.upper()</code> <code>cadena.split(...)</code> <code>cadena.find(...)</code> <code>cadena.count(...)</code>