

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN  
CCPG1001 - FUNDAMENTOS DE PROGRAMACIÓN  
SEGUNDA EVALUACIÓN - II TÉRMINO 2019-2020/ Enero 31, 2020

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 (10 PUNTOS)**

**¿Qué imprime el siguiente código? Justifique su respuesta**

```
a = {2, 8, 14, 19, 20}
b = {14, 16, 20, 4, 2, 1}
c = a.symmetric_difference(b)
resultado = (a - c).union(b - c)
print(resultado)
```

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

**TEMA 2 (45/30 PUNTOS)**

Dado el archivo **especies.csv** con información sobre el número de ejemplares de diferentes especies de animales en varios países, en el siguiente formato:

```
especie,Honduras,Australia,...
Elefante,0,0,...
Koala,0,10325,...
...
```

Escriba lo siguiente:

1. La función **crear\_diccionario(archivo, n)** que recibe el nombre del archivo con la información de las especies y un entero **n**. La función retorna un diccionario donde la clave es la especie y el valor es una lista de tuplas con los nombres de los países que tienen más de **n** ejemplares y la cantidad real de ejemplares para ese país. Ejemplo para **n=100**:

```
{'Tigre de Bengala': [('India', 183), ('Nepal', 178)],
 'Panda': [('China', 295), ('Tailandia', 170), ('Japon', 237)],
 ...
}
```

2. La función **mas\_popular(dicEspecie, especie)** que recibe el diccionario de especies del numeral anterior (1.) y el nombre de una especie. La función retorna el nombre del país que tiene el mayor número de ejemplares de esa especie.
3. La función **pais\_especie(dicEspecie)** que recibe el diccionario de especies. La función retorna un nuevo diccionario de especies por país, donde la clave es el país y el valor es una lista con los nombres de las especies que existen en ese país.
4. La función **especies\_en\_comun(dicPais, lista\_paises)** que recibe el diccionario de especies por país generado en el numeral anterior (3.) y una lista con varios nombres de países. La función retorna una tupla con los nombres de las especies comunes para todos los países de la lista.

### TEMA 3 (45|30 PUNTOS)

Dada la siguiente matriz **M** y listas:

		1998		...	2003		...	2019	
		Cantidad Turistas	Capacidad Hotelera	...	Cantidad Turistas	Capacidad Hotelera	...	Cantidad Turistas	Capacidad Hotelera
Costa	Guayaquil			...			...		
	...			...			...		
	Manta			...			...		
	...	...	...	...	...	...	...	...	...
Sierra	Quito			...			...		
	...	...	...	...	...	...	...	...	...
	Cuenca			...			...		
	...	...	...	...	...	...	...	...	...
Oriente	Macas			...			...		
	...			...			...		
	El Puyo			...			...		
	...			...			...		

costa = ['Guayaquil', ..., 'Manta',...]

sierra = ['Quito', ..., 'Cuenca', ...]

oriente = ['Macas', ..., 'El Puyo', ...]

años = [1998, ..., 2003, ..., 2017, ..., 2019]

Escriba el código de Python para calcular las siguientes estadísticas:

1. Cantidad total de turistas que ingresaron a las ciudades del país en el año 2017. **(int)**
2. La capacidad hotelera promedio para cada una de las ciudades de la sierra considerando todos los años. **(vector de float)**
3. El número de años en los que la cantidad de turistas fue mayor que la capacidad hotelera para la ciudad de Machala. **(int)**
4. Los años en la que la capacidad hotelera del país fue superior a 5000. **(vector o lista de int)**
5. El nombre de las tres ciudades del país que más turistas en total han tenido. **(vector o lista de str)**

#### TEMA 4 (30 PUNTOS) \*Solo para estudiantes de Computación, Telemática y Mecatrónica

Asuma que tiene una matriz **M** de  $m$  filas y  $n$  columnas, con valores entre 0 (totalmente sano) y 100 (totalmente enfermo) que representan el nivel de infección del Coronavirus en todos los sectores de una ciudad. Ejemplo:

0	23	37	58
0	100	94	18
21	56	42	17
11	3	68	77

Implemente lo siguiente:

1. **[10 puntos]** La función **probabilidad\_contagio(M, f, c)** que recibe la matriz de niveles, los índices de fila y columna de una celda de la matriz y retorna la probabilidad de contagio para esa ubicación. La probabilidad se calcula como el promedio de los niveles de infección de **todas las celdas vecinas a f,c**. La celda (f,c) **nunca estará en el borde de la matriz**.

Asuma que ya tiene implementada la función **vacunar(M, f, c, dosis)** que actualiza la matriz M disminuyendo los niveles de infección en varias de sus celdas.

2. **[20 puntos]** Un programa principal que repita los siguientes pasos hasta que las probabilidades de contagio para las celdas (1,1) y (1, n - 1) de la matriz **M** sean menor a 20 cada uno:
  - A) Obtenga, de la matriz de niveles, los índices de fila y columna de la celda más infectada
  - B) Genere aleatoriamente un **valor entero par** de dosis entre 2 y 200
  - C) Usando la función **vacunar**, aplique la dosis del paso B) en la celda del paso A)

---//---

#### Cheat Sheet. Funciones y propiedades de referencia en Python.

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