

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN  
CCPG1001 - FUNDAMENTOS DE PROGRAMACIÓN  
PRIMERA EVALUACIÓN - I TÉRMINO 2019-2020/ Julio 5, 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 (20 PUNTOS)**

Escriba un programa en Python que implemente el "Juego de las Ruedas". Para esto genere aleatoriamente una lista de 15 elementos donde cuatro elementos deben decir "Rueda" y los otros once, "X".

Luego el programa deberá pedirle al jugador que ingrese por teclado índices entre 0 y 14 (**validar**). Asuma que el jugador siempre ingresa índices distintos. Si el índice ingresado por el usuario corresponde al de una "Rueda", gana \$1000. El jugador tiene **siete intentos** para hallar las cuatro "Ruedas". En cada intento muestre por pantalla el número total de "Ruedas" encontradas hasta el momento. Si el jugador encuentra las cuatro "Ruedas" se gana un carro. **El juego termina cuando encuentra las cuatro "Ruedas" o ha usado todos los intentos.**

Al final muestre el premio que recibe el jugador (cantidad de dólares o la palabra "carro" si encontró las cuatro ruedas).

**TEMA 2 (20 PUNTOS)**

Asuma que tiene una lista con el consumo de datos de las aplicaciones en su celular durante un mes como en el siguiente ejemplo:

```
junio = ['Facebook-10Megas', 'Spotify-55Megas', 'Spotify-112Megas', 'Whatsapp-12Megas', ...]
```

Observe que las aplicaciones en la lista se pueden repetir.

Muestre por pantalla la cantidad de "Megas" que ha utilizado durante el mes por cada aplicación.

### TEMA 3 (50 PUNTOS)

La fábrica ACME se encarga de fabricar diversos componentes para vehículos. Para esto, tiene varias líneas de fabricación. Asuma que tiene los siguientes arreglos:

```
P = array(["ct-32", "mto-990", "ct-32"...])
PF = np.array([789, 1500, 900 ...])
PD = np.array([26, 35, 70, ...])
D = np.array([300, 12, 100, ...])
CF = np.array([1000, 2000, 1100, ...])
```

Donde **P** es un arreglo con los códigos de las piezas fabricadas. El arreglo **PF** tiene la cantidad total de piezas fabricadas en el día. El arreglo **PD** indica la cantidad de piezas defectuosas de cada producto por día. El arreglo **D** indica el día del año de fabricación de la pieza (1 a 365). Finalmente, el arreglo **CF** indica cual es la capacidad máxima de productos que se pueden fabricar, para cada tipo de pieza, en un día determinado. **Un mismo producto se puede fabricar varias veces al año pero no varias veces en un mismo día.**

1. **[6 puntos]** Escriba una función llamada **produccionAnual(codigo, P, PF)** que retorne la cantidad total de piezas **codigo** fabricadas en el año.
2. **[12 puntos]** El rendimiento de fabricación de un producto se mide de la siguiente forma:

$$\text{rendimiento} = (\text{total\_piezas\_fabricadas} - \text{total\_piezas\_defectuosas}) / \text{total\_capacidad}$$

Escriba la función **rendimientoPromedio(codigo, P, PF, PD, CF, D)** que retorna el rendimiento promedio diario para **codigo**. El rendimiento promedio diario es igual a  $\text{rendimiento} / \text{total\_dias\_fabricacion}$ .

3. **[12 puntos]** Escriba la función **porcentajeAnualDefecto(codigo, P, PD, PF)**, que retorna el porcentaje de defectos que tiene un **codigo**. El porcentaje de defectos se calcula como sigue:

$$\text{porcentaje\_defectos} = (\text{total\_piezas\_defectuosas} / \text{total\_piezas\_fabricadas}) * 100$$

4. **[10 puntos]** Escriba la función **productosDefectuosos(codigos, P, PF, PD, porcentaje)**, que retorna un arreglo con los códigos de los productos que están en la lista **codigos** y que tengan un porcentaje de defectos mayor a **porcentaje**.
5. **[10 puntos]** Escriba la función **minimoPorcentajeDefecto(P, PF, PD)**, que retorna el código del producto con el menor porcentaje anual de defectos.

## TEMA 4 (10 PUNTOS)

¿Qué imprime el siguiente código? Justifique su respuesta

```
import numpy as np

arr = np.array([3,2,1,5,4,2,4,1])
d = "eaiuo"
p = '*'
for a in range(arr[arr <= 3].size):
    p = p + d[a] * (arr == a).sum() + '\n'
print(p)
```

**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 <b>arreglos</b> :	para <b>listas</b> :	para <b>cadena</b> s:	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>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> <code>cadena.replace(a,b)</code>	<code>rnd.randint()</code> <code>rnd.choice(lista)</code> <code>rnd.sample(lista,cant)</code> <code>rnd.shuffle(lista)</code>