**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**
**CCPG1001 - FUNDAMENTOS DE PROGRAMACIÓN**
**PRIMERA EVALUACIÓN - II TÉRMINO 2017-2018/ Diciembre 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 (20 PUNTOS)

Usted ha sido contratado como nuevo administrador en el taller de Papa Noél, en el Polo Norte. Como es una época muy agitada, existen muchas tareas por realizar, dadas en una lista llamada **tareas:**

**tareas = [‘pintar soldados’, ‘hornear galletas’, ‘armar muñecos’, ‘cortar papel de regalo’, … ]**

Para cada tarea, usted tiene el tiempo de inicio en minuto del día, y la duración de la tarea, también en minutos:

**inicio = [678, 200, 240, 423, …]**

**duración = [300, 800, 456, 112, …]**

Por ejemplo, la tarea ‘pintar soldados’ empieza en el minuto 678 del día, y tiene una duración de 300 minutos.

Su trabajo es planificar la mayor cantidad de tareas que se pueden realizar **en un día** de 1440 minutos. **Para ello seleccionará las tareas basadas en sus tiempos de finalización prefiriendo las tareas que terminan más temprano en el día**. Recuerde que la finalización de la tarea se calcula como el minuto de inicio + los minutos de duración. **Asuma que no existen tareas que finalizan en el mismo minuto.** Como en el taller trabajan muchos duendes, no es problema planificar tareas que se deben ejecutar simultáneamente.

En el ejemplo anterior, la tarea ‘pintar soldados’ termina en el minuto 978 (678 + 300) del día. Recuerde que no es necesario que se planifiquen todas las tareas en un mismo día.

Escriba un programa que nos diga el orden en que se tiene que ejecutar las tareas, y cuáles son. Su programa debe mostrar las tareas en orden de ejecución. Por ejemplo:

```
+--------------+
|Tareas del día|
+--------------+
1. Cortar papel de regalos
2. Vestir muñecas
3. …
```

**TEMA 2 (30 PUNTOS)**

Como asistente de médico, usted tiene la tarea de generar un informe de indicadores a partir de un examen de sangre. El resultado del examen se lo entregan como una cadena de texto. Los indicadores los puede identificar porque estos siempre estarán en **mayúsculas**, por ejemplo INR, WBC, RBC, TA, etc. Todo indicador va seguido de un espacio, luego un número con decimales, seguido de otro espacio en blanco y finalmente las unidades. Al final del resultado se encuentra el nombre del médico. Ejemplos de resultados:

```
resultado = "Resultado de Laboratorio 'Su Salud' Nombre del paciente: Jose Aimas E-mail del paciente: jose.aimas@gmail.com Resultados del laboratorio: INR 1.25 segundos BGT 180.12 mmol/dL HGB 13 g/dL ESR 3.2 mm/hora RBC 4000024.2 cel/uL TA 1.5 ng/dL WBC 123233.23 cel/uL. Los valores de este informe no representan un diagnóstico.  Firma médico responsable Dr. Juan Pozo"
```

```
resultado = "Resultado de Laboratorio 'Sana' Nombre del paciente: Ginger Irene Cruz Jurado Edad: 25 años E-mail: giircrju@espol.edu.ec Resultados: Azúcar BGT 180.12 mmol/dL Hemoglobina HGB 13 g/dL Hormonal TA 1.5 ng/dL.  Médico responsable Dra. Karina Elizabeth Plaza"
```

**La cantidad de indicadores puede variar. Los puntos no solo aparecen en los decimales, sino también para separar párrafos o en otras ocasiones como las direcciones de e-mail.**

Escriba un programa que nos muestre la información desglosada, el nombre del médico y una recomendación de si el paciente debe ir al endocrinólogo. Un paciente debe ir al endocrinólogo si su nivel de azúcar (BGT), está por encima de los 150 mmol/dL. **En caso de dar la recomendación, mostrar doble asterisco en el indicador BGT y la recomendación al final.** Para el primer ejemplo de arriba el informe sería:

```
INFORME DE LABORATORIO
**********************

INR                   1.25  segundos
BGT             180.12      mmol/dL **
HGB                 13      g/dL
ESR                3.2      mm/hora
RBC          4000024.2       cel/uL
TA                  1.5      ng/dL
WBC          123233.23       cel/uL

Médico: Juan Pozo
**Su nivel de azúcar es alto, se recomienda ir al endocrinólogo.
```

**No es necesario presentar el informe en el formato arriba descrito pero si lo hace, obtendrá 2 puntos extras en el examen.**

**TEMA 3 (40 PUNTOS)**

Asuma que tiene la siguiente matriz **M** con datos de ventas anuales en galones de las gasolineras de todo el país:

| | **Primax Alborada** | **PS Los Ríos** | **Mobil Cumbayá** | **...** | **Lutexsa CIA Ltda** | **PS Remigio Crespo** |
|---|---|---|---|---|---|---|
| **Regular** | 239034 | 678493 | 896321 | ... | 32438 | 554213 |
| **Extra** | 4568321 | 6745634 | 9754008 | ... | 3242342 | 3456123 |
| **Súper** | 234773 | 56743 | 123678 | ... | 4783 | 90874 |
| **...** | ... | ... | ... | ... | ... | ... |
| **Premium** | 45672 | 45212 | 90781 | ... | 3904 | 90431 |

También tiene los siguientes datos como vectores de Numpy:

```
tipoGasolina = np.array([‘Regular’, ‘Extra’, ‘Súper’, … ‘Premium’])
gasolineras = np.array([‘Primax Alborada’, ‘PS Los Ríos’, ‘Mobil Cumbayá’, …, ‘Lutexsa
CIA Ltda’,‘PS Remigio Crespo’])
distrito = np.array(['distrito1', 'distrito2', 'distrito1', … , ‘distrito2’,
'distrito4'])
ciudades = np.array(['Guayaquil', 'Babahoyo' , 'Quito'    , … , ‘Guayaquil’, 'Cuenca'])
```

El vector **tipoGasolina** contiene todos los nombres de los diferentes tipos de gasolina que se comercializan en el país. Los siguientes tres vectores, **gasolineras**, **distrito** y **ciudades**, contienen el nombre de la estación, el distrito y la ciudad en la que se encuentra cada estación. Cada **gasolinera** está relacionada con su respectivo **distrito** dentro de una **ciudad**. Los nombres de los distritos y de las ciudades se pueden repetir dentro de sus respectivos vectores.

Implemente código en Python para los siguientes problemas:

1. **[12 puntos]** Pida un tipo de gasolina por teclado y muestre por pantalla los **nombres** de todas las gasolineras que han vendido en el año **más del promedio de venta** en galones para ese tipo.
2. **[13 puntos]** Pida una ciudad por teclado y calcule cuántas de sus gasolineras han vendido **más de 15 millones** de galones en total en el año, considerando todas las ventas de todos los tipos de gasolinas.
3. **[20 puntos]** Muestre por pantalla el nombre de la ciudad que más galones ha vendido en el año de gasolina tipo **Diesel** en el distrito **distrito1**.

**TEMA 4 (10 PUNTOS)**

1. Indique la salida por pantalla del siguiente código. Justifique su respuesta.

```
f = ['a', 'c', 'z', 'm', 'k']
g = [3,4,5,6,5,7]
t = ''

for c in f:
   a = f.index(c)
   b = g[:a]
   t = t + (c * len(b))

print(t)
```

2. Indique la salida por pantalla del siguiente código. Justifique su respuesta.

```
import numpy as np
vector = np.array([1, 5, 6, 6, 5, 2, 1, 3, 7, 9, 0, 0, 1, 4, 8])
print(np.unique(vector[vector % 2 == 0]).size)
```

---//---

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

| Librería Numpy para ***arreglos*:** | para ***listas***: | para ***cadenas***: |
|---|---|---|
| np.array((numRows,numCols),dtype= )<br>***arreglos***.shape<br>***arreglos***.reshape()<br>np.sum(***arreglos***)<br>np.mean(***arreglos***)<br>np.where(**condición**)<br>np.unique(**arreglo**)<br>***arreglos***.sum(axis=1) | ***listas***.append(...)<br>***listas***.count(...)<br>***listas***.index(...)<br>***listas***.pop()<br>*elemento* in ***listas*** | ***cadenas***.islower()<br>***cadenas***.isupper()<br>***cadenas***.lower()<br>***cadenas***.upper()<br>***cadenas***.split(...)<br>***cadenas***.find(...)<br>***cadenas***.count(...) |