

MEGAS DE FUNDAMENTOS DE PROGRAMACIÓN

EJERCICIO 1. Escriba un programa para obtener una cadena de caracteres dada donde todas las ocurrencias de su primer carácter han sido cambiadas a '\$' excepto el primer carácter. No importa si la palabra esta en mayúscula o minúscula, igual se debe cumplir la condición. **Ejemplo:**

Ingrese palabra: REPARAR

Ingrese palabra: Lapilla

Palabra resultante: REPA\$A\$

Palabra resultante: LAPI\$A

EJERCICIO 2. Escriba un programa en Python que determina si una matrícula de novato es válida. Para que la matrícula sea válida debe cumplir con los siguientes requisitos:

- La matrícula debe comenzar con 2019
- Debe de contener 9 números.

Ejemplo:

Ingrese una Matricula: 201611882

La matrícula 201611882 no es válida...

Ingrese una Matricula: 201923457

La matrícula 201923457 es válida !

EJERCICIO 3. Usted cuenta con 3 listas: matricula, nombres y promedios:

```
matricula= ["201611882", "201923457", "201957410", ...]  
nombres= ["Villamar Saltos", "Santos Briones", "Roca Villarroel", ...]  
promedio= [ 78, 80, 54, ...]
```

OJO: El estudiante con matrícula 201923457 es Santos Briones y aprobó con un promedio de 80. Escriba un programa que permita el ingreso de una matrícula, el programa mostrará el apellido y el promedio si el estudiante aprobó, si el estudiante no aprobó el programa mostrará que reprobó la materia. (hacer las validaciones correspondientes, hasta que se ingrese una matrícula existente)

Ejemplo 1:

Ingrese matricula: 201923000

Esa matricula no existe.....

Ingrese matricula: 201923457

Santos Briones usted aprobó la materia con un promedio de: 80

Ejemplo 2:

Ingrese matricula: 201957410

Roca Villarroel usted reprobó la materia.

EJERCICIO 4. 1ra Evaluación II Término 2016

Las aplicaciones para análisis de texto en internet (robots) siempre están trabajando para mejorar la eficacia del reconocimiento lingüístico. Para esta misión, los robots investigan el alfabeto teniendo como referencia:

```
vocales = 'AEIOU'  
consonantes = 'BCDFGHJKLMNPQRSTVWXYZ'
```

Escriba un programa que dado un bloque de texto con palabras separadas por un espacio (' ') o un punto ('.'), Encuentre el número de palabras que tienen la misma cantidad de vocales y consonantes.

Ejemplos:

Texto:	Resultado
Mi nombre eS	2
Hola mundo	1
H0la mundo	0
Algunas Palabras pArA contar en un programa con algo de Python	5
Perro.gato.raton.pajaro.Humano	3

Considere que el texto:

- No contiene vocales con tildes.
- NO se encuentran dos o más espacios seguidos, pero si combinaciones de espacios y puntos.
- Las mayúsculas y minúsculas no afectarán el resultado
- Una palabra que contenga números no se la analiza.

EJERCICIO 5. 1ra Evaluación I Término 2017

En el juego de tablero “Scrabble”, cuando el jugador forma una palabra, acumula los puntos de los equivalentes para cada letra.

Una palabra se puede crear cruzando otra palabra y usando una letra compartida que recibe el doble de puntos.

```
alfabeto = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
equivale = [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]
```

Realice un programa en que el jugador ingrese varias palabras separadas por comas (','), determine el puntaje correspondiente a cada palabra y muestre cuál es la palabra con mayor puntaje.

```
Ejemplo:
>>>
Analizar: CAS*A*,S*ASTR*E*,R*EY*,A*ZOTE*
CAS*A* 8
S*ASTR*E* 9
R*EY* 11
A*ZOTE* 16
Ganador: A*ZOTE* 16
```

Nota: Para marcar una letra compartida se añade un símbolo ('*'), suponga que todas las palabras ingresadas terminan con una letra compartida. Todas las letras deben ser mayúsculas, si se ingresa una minúscula será ignorada para el puntaje.

EJERCICIO 6. 1ra Evaluación II Término 2016

Se dispone del registro total por día de las transacciones de las tiendas deportivas, Marathon Sports en Guayaquil. Cada registro comprende un texto que representa los valores de:

```
sector|tienda|categoria|producto|totalVentas|dia-mes-año
```

Ejemplo:

```
transacciones = [  
    'centro|Bahia|futbol|zapatos-Adidas|290.78|25-03-2013',  
    'centro|Malecon2000|natacion|chaleco-Fins|110.92|01-02-2014',  
    'sur|MallDelSur|natacion|gafasPiscina-Swingo|90.07|13-05-2014',  
    'centro|Bahia|natacion|zapatos-Nike|315.72|13-12-2015',  
    'norte|CityMall|natacion|gafasPiscina-Adidas|310.19|31-05-2016',  
    '...',  
]
```

Escriba un programa que usando los datos de transacciones encuentre:

1. Los nombres de las tiendas que pertenecen a cada sector (sur, centro, norte). Considere usar una lista por cada sector con nombres de tiendas no repetidos.
2. El total de ventas de los productos 'Adidas' en el mes de mayo ('05') del año que el usuario requiera.

Nota: Considere desarrollar el tema de forma independiente del anterior.

EJERCICIO 7.- 1ra Evaluación I Término 2019

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

```
juno = ['Facebook-10MB', 'Spotify-55MB', 'Spotify-112MB', 'Whatsapp-12MB', ...]
```

Observe que las aplicaciones en la lista se pueden repetir.

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

EJERCICIO 8. 3ra Evaluación I Término 2019

Que imprimen los siguientes códigos

```
A = 'Num empresas,17,0|9|1,10|19|2,20|29|3'
campos = A.split(' ')
valores = []
for rango in campos[2:]:
    valores.append(int(rango.split('|')[0]))
print(valores)
```

```
pal = 'Se van en sus naves'
b = pal[::-1].replace(' ', '').lower()
pal_b = pal.lower().replace(' ', '')
if pal_b == b:
    print('Es palíndromo')
else:
    print('No es palíndromo')
```

EJERCICIO 9. 1ra Evaluación I Término 2017

La lista mostrada en el ejemplo contiene los URLs de diferentes sitios Web que han sido visitados.

```
lista = ["www.espol.edu.ec",
        "www.google.com",
        "www.sri.gob.ec",
        "www.fiec.espol.edu.ec",
        "www.uess.edu.ec",
        "www.FIEC.espol.edu.ec",
        "www.fict.espol.edu.ec",
        "www.fcnm.Espol.edu.ec",
        "www.ucsg.edu.ec",
        "www.Stanford.edu",
        "www.harvard.edu",
        "www.stanford.edu",
        "www.UCSG.edu.ec",
        "www.google.com.ec",
        "www.facebook.com",
        "www.opensource.org",
        "www.educacionbc.edu.mx",
        "..."]
```

Los URLs normalmente se repiten y corresponden algunas veces a universidades de Ecuador y otros países.

Note que los URLs no diferencian entre mayúsculas y minúsculas.

Por ejemplo: `www.espol.edu.ec` y `www.ESPOL.edu.ec` corresponden al mismo sitio.

Escriba un programa en **Python** que dada una lista realice lo siguiente:

Se le pide:

1) Escriba una función llamada **universidadesUnicas()** que recibe una lista de las URLs de las universidades y retorna una lista con las siglas o nombres de las universidades sin repetir en mayúscula. Ejemplo de lo que debe retornar la función:

```
["ESPOL", "UESS", "UCSG", "STANFORD", "HARVARD", "EDUCACIONBC"]
```

2) Escriba una función llamada **universidadesUnicasEcuador()** que recibe una lista de las URLs de las universidades y retorna una lista con las siglas o nombres de las universidades sin repetir en mayúscula del ecuador. Ejemplo de lo que debe retornar la función:

```
["ESPOL", "UESS", "UCSG"]
```

MEGAS DE FUNDAMENTOS DE PROGRAMACIÓN

EJERCICIO 10. 1ra Evaluación I Término 2016

Una empresa registra los sitios de internet visitados por sus empleados y los minutos de navegación en una lista de tamaño t semejante a la mostrada.

Cada registro se almacena usando una cadena con el formato:

```
visitados = [ 'maria2|www.facebook.com|160',  
              'xavi7|www.eluniverso.com|50',  
              'jose15|www.sri.gob.ec|30',  
              'maria2|www.twitter.com|30',  
              'xavi7|www.inec.gob.ec|10',  
              'maria2|www.espol.edu.ec|50',  
              'jose15|www.sri.gob.ec|120',  
              'xavi7|www.sri.gob.ec|20',  
              'maria2|www.twitter.com|20',  
              '...' ]
```

Para analizar los datos, la empresa dispone de las listas para n empleados y p sitios de trabajo:

```
empleados = ['maria2', 'jose15', 'xavi7', '...']  
trabajo = [  
    'www.espol.edu.ec',  
    'www.inec.gob.ec',  
    'www.sri.gob.ec',  
    '...']
```

Con las listas `visitados`, `empleados` y `trabajo` descritas, elabore un programa en Python para obtener los siguientes reportes:

- Genere una lista con los sitios que no son de trabajo que han sido visitados por los empleados. La lista obtenida debe contener los q los elementos sin repetir.
- El nombre del usuario que más tiempo ha pasado en sitios de no trabajos

EJERCICIO 11. 1ra Evaluación II Término 2018

Asuma que la lista **palabras** contiene un número determinado de palabras que se usarán para al juego “Conocimiento” que le han pedido implementar basado en la siguiente mecánica:

- La computadora selecciona aleatoriamente una palabra con la que se jugará de la lista **palabras**. Esta palabra deberá ser almacenada en la variable **secreto**.
- La computadora escogerá una consonante de **secreto** aleatoriamente y, asumiendo que **secreto** es “ESPOL” y la letra seleccionada es la ‘s’, mostrará por pantalla lo siguiente:

-S---

- El usuario tendrá 2 x (el tamaño de la palabra almacenada en **secreto**) turnos, para adivinar el resto de la palabra.
- El usuario puede ingresar letras mayúsculas o minúsculas y esto no debe hacer diferencia en el juego.
- Si la letra ingresada pertenece a la palabra en **secreto**, se mostrará por pantalla la nueva letra adivinada en la posición que le corresponde.

Escriba un programa de Python que implemente el juego y muestre por pantalla si el usuario ganó o perdió y la cantidad de turnos que empleó.

EJERCICIO 12.- 1ra Evaluación I Término 2019

Escriba un programa en Python que implemente el “Juego de las Ruedas”. Para esto genere **aleatoriamente** una lista de 12 elementos donde cuatro elementos deben decir “Rueda” y los otros ocho deben decir “X”.

Luego el programa deberá pedirle al jugador que ingrese por teclado índices entre 0 y 11 (**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 **seis 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).

EJERCICIO 13. 1ra Evaluación II Término 2017

Usted ha sido contratado como nuevo administrador en el taller de Papa Noel, 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. ...
```

EJERCICIO 14. Pruebas de escritorios años 2015 - 2016

```
lista1 =[3, 'A', 6]
lista2 =['A']

def funcion (lista1, lista2):
    a=[]
    for i in lista1:
        for j in lista2:
            if i != j:
                a.append(str(i) + str(j))
                for x in a[:]:
                    a.append(str(i) + str(j))
    return(a)

print(funcion (lista1, lista2))
```

```
a = 'ABACDEF'
b = [4,2,1,3,2,2,3]
d = ''
for i in a:
    if i not in d:
        d += i*b[ list(a).index(i)]
    else:
        d = d.replace(i, '')
print(d)
```

```
a = [1,3,5,6,8]
b = []
for i in range(len(a)):
    b.append(i)
    b.insert(i, len(b))
print(b)
```


MEGAS DE FUNDAMENTOS DE PROGRAMACIÓN

EJERCICIO 15. 1ra Evaluación II Término 2017

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.

EJERCICIO 16. 1ra Evaluación II Término 2015

- Al inicio, todos los jugadores ubican su ficha en la casilla 1 con \$1500.
- El tablero es de recorrido cíclico de **24 casillas**.
- Se juega por turnos avanzando con la suma del lanzamiento de **dos dados**.
- Un jugador al caer en las casillas de “**Casualidad**” (4, 10, 16) debe pagar \$100 a otro jugador escogido aleatoriamente.
- Cada jugador tiene cinco casillas como “**propiedades**”; si un jugador cae en la propiedad de otro le deberá pagar del derecho de ocupación un valor determinado al lanzar un dado y multiplicarlo por 10.



Considere que:

- El jugador 1 es propietario de las casillas 2, 8, 13, 19, 22.
- El jugador 2 es propietario de las casillas 3, 7, 11, 17 y 24.
- El jugador 3 es propietario de las casillas 5, 9, 15, 20 y 23.
- El jugador 4 es propietario de las casillas 6, 12, 14, 18 y 21.
- Cada vez que un jugador pase por “**Inicio**” recibirá \$200.

Realice un algoritmo que simule el juego hasta que **uno de los jugadores quede en bancarrota** (sin dinero). Al final, indique el jugador que se quedó en la bancarrota y el dinero que obtuvo cada jugador.

EJERCICIO 17. 1ra Evaluación I Término 2014

Tema 2. (25 puntos) El número **EAN-13** (European Article Number) usado comercialmente en Europa en la identificación de productos, está constituido por 13 dígitos y con una estructura dividida en cuatro partes :

- 3 dígitos para el país,
- 4 dígitos para la empresa,
- 5 dígitos para el producto,
- y
- un dígito de control.



El dígito de control permite detectar errores de lectura del código, calculado como:

- Comenzando por la derecha, se multiplican los dígitos del código por 1 si su **posición es par** y por 3 si es **impar**,
- Se suman los valores de los productos obtenidos,
- Se resta a la decena superior el resultado de la suma, siendo el resultado el dígito de control.

7	7	0	2	0	0	4	0	0	3	5	0
x1	x3	x1	x3	x1	x3	x1	x3	x1	x3	x1	x3
7	21	0	6	0	0	4	0	0	9	5	0

```
Suma = 52
decena superior = 60
verificador calculado 60-52 = 8
```

Realice un algoritmo que dado un número EAN valide que sea de 13 dígitos, calcule el dígito verificador e informe si es correcto.