

Examen IP 2023 IT

Solución: Xavier Pauta

Tema 1 (15 puntos)

Implemente la función `bucar_palindromo(mensaje)` que recibe un mensaje y retorna una lista con todas las palabras (**de 3 o mas letras**) del mensaje que son palindromos. Asuma que todas las palabras estan en minusculas y no existen signos de puntuacion.

Nota: Recuerde que un palindromo es una cadena que se lee igual de izquierda a derecha o de derecha a izquierda. Por ejemplo: madam, ana, somos, reconocer, anilina.

Ejemplo de entrada

```
mensaje= 'ana y yo sono amigos y trabajamos en la torre del radar'
```

Ejemplo de salida

```
['ana', 'somos', 'radar']
```

Solucion:

```
def buscar_palindromos(mensaje):
    palabras= mensaje.split(' ')
    palindromos=[]
    for palabra in palabras:
        if palabra==palabra[::-1] and len(palabra) ≥ 3:
            palindromos.append(palabras)
    return palabras
```



Tema 2 (15 puntos)

Implemente un programa que:

1. Pida una palabra al usuario
2. Pida al usuario una frase a la vez hasta que este ingrese la palabra 'Final'.
3. Elimine de cada frase las comas y los puntos. Asuma que las frases no tienen ningun otro sino de puntuacion.
4. AL finalizar, muestre por pantalla cuantas veces se repitio la palabra ingresada en el paso 1 en las frases ingresadas en el paso 2. Al momento de contar asegurese de considerar mayusculas y minusculas para que cuenten como la misma palabra. Ejemplo: 'hola' y 'Hola' son la misma palabra

Asegurese de comparar palabras exactas. Ejemplo: 'Hola' en 'Estoy en Holanda' no cuenta.

Ejemplo de entrada:

```
Ingrese una palabra: hola
Ingrese una frase (Ingrese 'Final' para terminar): Hola, estoy bien
Ingrese una frase (Ingrese 'Final' para terminar): Hola. No me gusta saludar diciendo hola
Ingrese una frase (Ingrese 'Final' para terminar): Final
```

Ejemplo de salida:

La palabra 'hola' se repitio 3 veces en las frases ingresadas.

Solucion:

```
palabra=input('Ingrese una palabra: ').lower()
palabras= []
frase=''
total=0
while frase!='final':
    frase= input('Ingrese una frase: ').lower()
    frase= frase.replace(',','').replace('.', '')
    if frase!='final':
        frase= frase.split(' ')
        total+= frase.count(palabra)
print('Total de veces que se repite la palabra {} en cada frase: ', total)
```



Tema 3 (25 puntos)

Implemente la función `siglas(frase, palabras_comunes)` que recibe una frase y una lista de palabras comunes. Asuma que tanto las palabras de la frase como de la lista están en minúsculas. La función debe generar una sigla utilizando las iniciales en mayúsculas de cada palabra. La función debe omitir de la sigla palabras comunes como 'de', 'y', 'en', etc. (definidas en el parámetro `palabras_comunes`), a menos que sean la primera o la última palabra de la frase. Asuma que en la frase no existen signos de puntuación.

Ejemplo de entrada

```
frase= 'el yin y el yang son conceptos filosoficos de la cultura china'
palabras_comunes= ['el', 'y', 'de', 'la', 'son']
```

Ejemplo de salida

```
'EYYCFCC'
```

Solución:

```
def siglas(frase, palabras_comunes):
    siglas=''
    lst_frase= frase.split(' ')
    for i in range(len(lst_frase)):
        palabra= lst_frase[i]
        sigla= palabra[0].upper()
        if i==0 or i==len(lst_frase)-1:
            siglas+= sigla
        else:
            if palabra not in palabras_comunes:
                siglas+= sigla
    return siglas
```



Tema 4 (25 puntos)

Implemente a funcion `rotacion_parcial(numeros, k)` que recibe como parametros una lista de numeros y un numero entero positivo `k` mayor a cero y menor que el tamaño de la lista. La funcion realiza una rotacion parcial hacia la izquierda de la lista, pero solo para los primeros `k` elementos y retorna una nueva lista

Ejemplo de entrada

```
lista = [110, 202, 343, 40, 52, 657, 743, 81]
```

Ejemplo de salida

```
[343, 110, 202, 40, 52, 657, 743, 81]
```

Explicacion Para el ejemplo de `k=3` solo los tres primeros elementos de la lista cambian su orden. El resto de elementos de la lista mantienen su orden original.

```
|-----k=3-----|
[110, 202, 343, 40, 52, 657, 743, 81]
|      |--->|
|      |    |
|--->|    |
|      |    |
|<-----
[343, 110, 202, 40, 20,
```

Solucion

```
def rotacion_parcial(numeros, k):
    L= numeros.copy()
    k_elemento= L.pop(k-1)
    L.insert(0, k_elemento)
    return L
```



Tema 5

Implemente la función `buscar_divisibles(numeros, divisor)` que recibe como primer parametro una lista con numeros enteros y como segundo parametro un numero entero positivo mayor a 0. La función retorna una nueva lista con los numeros que son divisibles para el numero recibido en el segundo parametro.

Luego, implemente un programa principal y realice lo siguiente

1. Genere una lista con 32 numeros aleatorios entre 5 y 42
2. Genere un numero aleatorio entre 6 y 19
3. Llame a la función con la lista y numero generados en los pasos anteriores y muestre la lista resultante

Ejemplo de entrada

```
numeros_aleatorios = [228, 213, 277, 169, 230, 168, 110, 225, 58, 147,  
                      104, 322, 249, 164, 178, 162, 192, 50, 61, 183,  
                      198, 326, 102, 210, 90, 334, 30, 40, 331, 172,  
                      337, 299]  
  
divisor= 9
```

Ejemplo de salida

```
[225, 162, 198, 80]
```

Solucion

```
import random as rd  
def buscar_divisibles(numeros, divisor):  
    divisibles=[]  
    for numero in numeros:  
        if numero%divisor==0:  
            divisibles.append(numero)  
    return divisibles  
  
aleatorios= []  
while len(aleatorios)<32:  
    numero= rd.randint(5, 42)  
    if numero not in aleatorios:  
        aleatorios.append(numero)  
  
divisor= rd.randint(6, 19)  
result= buscar_divisibles(aleatorios, divisor)  
print(result)
```

