

Instituto Tecnológico de Costa Rica

Escuela de Computación

Introducción a la Programación / Taller de programación.

Profesor: Ing. Jeff Schmidt Peralta

## EJERCICIOS SOBRE PROGRAMACIÓN RECURSIVA PARA 1er PARCIAL.

1. Escriba una función recursiva de pila `divida(dig, lista)` que reciba un dígito y una lista y obtenga dos listas, la primera compuesta por los dígitos mayores o iguales al dígito dado y la segunda compuesta por los dígitos menores al dígito dado. Sugerencia: utilizar funciones lambda. La función debe comportarse como los siguientes ejemplos:

```
>>> divida(5, [1, 0, 2, 9, 0, 9, 9])    >>> divida(8, [1, 0, 2, 6, 0])
([9, 9, 9], [1, 0, 2, 0])              ([], [1, 0, 2, 6, 0])
>>> divida(5, [8, 1, 5, 2, 6, 0])      >>> divida(7, [8])
([8, 5], [1, 2, 6, 0])                 ([8], [])
```

2. En una competencia de talentos 10 jueces califican a los competidores en una escala de 1 a 100. Para obtener la calificación de un competidor, se eliminan la más alta y la más baja y se hace un promedio de las calificaciones restantes. Debe implementar usando recursividad la función Python `calificacion(lista)` que recibe una lista con las calificaciones de los jueces y que obtenga la calificación de un competidor.

```
>>> calificacion([6, 8, 8, 8, 8, 8, 8, 10, 8, 8])
8
```

3. Escriba una función en Python a llamar `intercambiado(num)` que reciba un número entero e indique si cada uno de los dígitos que lo componen se intercambian entre pares e impares, considerando el cero como par. La función debe comportarse de la siguiente forma:

```
>>> intercambiado(943652)
True
>>> intercambiado(3423830)
False
```

4. Escriba una función recursiva de pila `cuenta(dig, num)` que reciba un dígito y un número entero y obtenga dos contadores, el primero compuesto por la cantidad de dígitos que son divididos en forma exacta por el dígito dado y el segundo compuesto por la cantidad de dígitos que no son divididos en forma exacta al dígito dado. La función debe comportarse como los siguientes ejemplos:

```
>>> cuenta(2,1029099)      >>> cuenta(3,396366)
(1, 6)                      (6, 0)
>>> cuenta(5,15)           >>> cuenta(7, 8)
(1, 1)                      (0, 1)
```

5. Escriba una función `parejas(num)` en la cual `num` es un número entero y que cuente las veces que aparecen dígitos en parejas. Ejemplos:

```
>>> parejas(41234426601)    >>> parejas(333)
2                             1
>>> parejas(234601)         >>> parejas(9991222)
0                             2
```

6. Escriba una función a llamar `cambiar(lista, ele)` que reciba una lista y un elemento y cambie por un cero la primera aparición de ese elemento en la lista.

```
>>> cambiar([0, 5, 1, 2, 5, 3, 4], 5)
[0, 0, 1, 2, 5, 3, 4]

>>> cambiar([0, 5, 1, 2, 5, 3, 4], 8)
[0, 5, 1, 2, 5, 3, 4]
```

7. Escriba una función a llamar `cambiar_todos(lista, ele)` que reciba una lista y un elemento y cambie por un cero todas las apariciones de ese elemento en la lista.

```
>>> cambiar_todos([0, 5, 1, 2, 5, 3, 4], 5)
[0, 0, 1, 2, 0, 3, 4]
>>> cambiar_todos([0, 5, 1, 2, 5, 3, 4], 8)
[0, 5, 1, 2, 5, 3, 4]
```

8. Se desea recibir una lista no nula y devolver otra compuesta por dos sublistas, la primera tendrá a los elementos pares, mientras que la segunda contendrá a los impares. Sugerencia: use `lambda`

```
>>> separa([12,14,16,20,30] )  >>> separa([12,1,20,3])
[[12,14,16,20,30], []]        [[12,20],[1,3]]
```