

EJERCICIOS RECURSIVIDAD DE COLA 5a.

1. Escriba una función `mayor(num)` que reciba un número entero y obtenga el dígito mayor. Escribir dos soluciones: una utilizando recursividad de pila y otra usando recursividad de cola. Haga una comparación de las soluciones y de las pruebas manuales.

```
>>> mayor(9978)          >>> mayor(1000009)
9                          9
```

2. Escriba una función recursiva en cola que reste un dígito específico a todos los dígitos de otro número. Si la resta es negativa se debe poner un cero. Por ejemplo:

```
>>> restard(7, 9978)
2201
>>> restard(7, 1084)
10
```

3. Escribir una función en recursividad de cola llamada `cambia`, que reciba un número entero y cambie los dígitos que sean un divisor de 4, por un cero, retornando un número entero.

```
>>> cambia(1488)          >>> cambia(72571)
1000                      72571
>>> cambia(4275)
275
```

4. Escriba una función llamada `hay_div` que recibe un número y un dígito y retorne `True` si al menos un dígito del número es divisor exacto del dígito recibido y `False` en caso contrario.

```
>>> hay_div(462148, 2)
True
>>> hay_div(13579, 2)
False
```

5. Escriba una función llamada `todos_div` que recibe un número y un dígito y retorne `True` si todos los dígitos del número son divisores exactos del dígito recibido y `False` en caso contrario.

```
>>> todos_div(46248, 2)
True
>>> todos_div(16248, 2)
False
```

6. Escriba una función `revise_num` que reciba un número (que debe ser entero) y retorne una tupla que tenga la siguiente forma:

(cantidad-dígitos-pares, cantidad-dígitos-impares)

La función debe retornar su resultado en forma similar al siguiente ejemplo:

```
>>> revise_num(482401)
(5, 1)
>>> revise_num(4)
(1, 0)
```

5. Escriba una función recursiva en cola `divida(dig, num)` que reciba un dígito y un número entero y obtenga dos números, el primero compuesto por los dígitos mayores o iguales al dígito dado y el segundo compuesto por los dígitos menores al dígito dado. La función debe comportarse como los siguientes ejemplos:

```
>>> divida(5, 1029099)      >>> divida(8, 10260)
(999, 1020)                (0, 10260)
>>> divida(5, 815260)      >>> divida(7, 8)
(85, 1260)                 (8, 0)
```

6. Escriba una función `intersec(num1, num2)` que recibe 2 números enteros y forma un nuevo número con los dígitos del primer número que están contenidos en el segundo número.

```
>>> intersec(24, 42)        >>> intersec(333, 3)
24                          333
>>> intersec(123, 692)      >>> intersec(435, 2340)
2                           43
```

7. Analice la siguiente serie numérica:

2, 5, 7, 10, 12, 15, 17, 20 ...

Determine su fórmula y construya una función recursiva en cola llamada `serie(n)` que reciba como argumento un número y retorne el número correspondiente de la serie al número dado.

```
>>> serie(5)          >>> serie(8)
12                     17
```

6. La siguiente fórmula define una sumatoria que converge a un valor:

$$\sum_{i=1}^N 1/i$$

Escriba una función usando recursividad de cola `sumatoria(n)` que reciba el límite superior de la sumatoria y calcule el resultado hasta ese número. Haga la prueba con diferentes valores de `n` para determinar hacia que valor converge.

8. Escriba una función en cola `multdig(num1, num2)` que recibe 2 números enteros del mismo tamaño y forma un nuevo número con la multiplicación de cada dígito del primer número con cada dígito del segundo número. Si la multiplicación de 2 dígitos es mayor a 9, se toma el dígito menos representativo del resultado de la multiplicación. Los siguientes ejemplos muestran como debe comportarse la ejecución de la función:

```
>>> multdig(24, 42)
88
>>> multdig(323, 388)
964
>>> multdig(153, 632)
656
```

9. Escriba una función `moda(num)` con recursividad de cola, que reciba un número entero y obtenga el dígito que más veces se repite.

```
>>> moda(9178)          >>> moda(1000009)
-1                       0
```