

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Introducción a la Programación / Taller de programación.

I semestre 2018

Profesor: Ing. Jeff Schmidt Peralta

Ejercicios sobre programación iterativa

1. Implementar la versión iterativa del algoritmo que calcula el factorial para un número n.

```
>>> factorial(6)
720
```

2. Implementar la versión iterativa del algoritmo que dado un valor para n encuentra el valor dentro de la sucesión de Fibonacci para ese n.

```
>>> fib(7)
13
>>> fib(35)
24157816
```

3. Escriba una función booleana iterativa `tiene_nueve(num)` que recibe un número entero y verifica si el número dado tiene al menos un dígito nueve. La función debe comportarse de la siguiente forma:

```
>>> tiene_nueve(900035)
True
>>> tiene_nueve(83735)
False
```

4. Un número es palíndromo si puede leerse igual de izquierda a derecha que de derecha a izquierda. Escribir una función usando iteración llamada `palindromo(num)` que reciba un número y retorne `True` si el número recibido es un palíndromo y `False` si no. Por ejemplo:

```
>>> palindromo(38583)
True
>>> palindromo(4)
True
>>> palindromo(2442)
True
>>> palindromo(1010010)
False
```

5. Escriba una función iterativa booleana `suma15(num)` que recibe un número entero y verifica si la suma de sus dígitos es mayor o igual a 15. La función debe comportarse de la siguiente forma:

```
>>> suma15(80642)
True
>>> suma15(2000412)
False
```

6. Escriba una función por medio de iteración `digitos` que reciba un número (que debe ser entero) y retorne una tupla que tenga la siguiente forma:

(cantidad-dígitos-mayores-que-5, cantidad-dígitos-menores-o-iguales-que-5)

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

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

7. La operación de multiplicar dos números $a * b$, cuando b es un número entero, puede calcularse por medio de $b - 1$ sumas sucesivas del número a . Por ejemplo la multiplicación $8 * 4$, podría verse como: $8 + 8 + 8 + 8$ (tres sumas sucesivas de 8). Escriba una función iterativa `multi(num1, num2)` que reciba dos números e implemente la multiplicación por medio de sumas sucesivas. La ejecución de la función debe mostrar resultados como los siguientes:

```
>>> multi(8, 4)
32
>>> multi(-9, 5)
-45
>>> multi(-9, -5)
45
```

8. Hacer una función iterativa `forma_par(num)` que recibe un número entero y forma otro número con los dígitos pares del número de entrada.

<pre>>>> forma_par(93235) 2 >>> forma_par(4444) 4444</pre>	<pre>>>> forma_par(423114) 424 >>> forma_par(113377) 0</pre>
--	--

9. Escriba una función iterativa `todos_pares(num)` que recibe un número entero y verifica si el número dado tiene todos sus dígitos pares. La función debe comportarse de la siguiente forma:

```
>>> todos_pares(80642)
True
>>> todos_pares(201462)
False
```

10. Escriba una función iterativa `todos_div(num, dig)` que recibe un número y un dígito y retorne `True` si todos los dígitos del número son divisibles entre el dígito recibido y `False` en caso contrario.

```
>>> todos_div(8462, 2)      >>> todos_div(42166, 2)
True                        False
```

11. Escriba una función iterativa `iguales(num)` que recibe un número entero y verifica si el dígito más significativo y el menos significativo son iguales. La función debe comportarse de la siguiente manera:

```
>>> iguales(93235)
False
>>> iguales(348383)
True
```

12. Escriba una función iterativa `elimine_n(dig, num, veces)` que recibe un dígito, un número entero y una cantidad de veces y elimina el dígito indicado en el número, la cantidad de veces que aparece. La función debe comportarse de la siguiente manera:

```
>>> elimine_n(3, 9303235, 2)
93025
>>> elimine_n(5, 348383, 2)
348383
>>> elimine_n(3, 348383, 5)
488
```