Instituto Tecnológico de Costa Rica

Escuela de Computación

Introducción a la Programación / Taller de programación.

Profesor: Jeff Schmidt Peralta I Semestre 2018

EJERCICIOS PARA II EXAMEN PARCIAL.

1. Escriba una función iterativa a llamar `iota(num)` que reciba un número natural y obtenga una lista en orden ascendente de números naturales menores al número dado.

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

2. Escriba una función iterativa llamada `dígitos(lista)` que reciba una lista de dígitos (debe verificarlo) y retorne una lista que tenga la siguiente forma:
   [cantidad-números-mayores-que-5, cantidad-números-menores-o-iguales-que-5]

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

3. En una lista se reciben las notas de n estudiantes de un curso. Escriba una función iterativa notas(lista) que considerando que la nota de aprobación sea 70 y obtenga:
   a) Cuántos estudiantes aprobaron el curso.
   b) Cuantos estudiantes reprobaron.
   c) El promedio de notas.

4. Escribir una función iterativa llamada `parcial(lista, inicial, final)` que reciba como argumentos una lista de números, una posición inicial y una posición final y obtenga una lista conteniendo los elementos encontrados entre las posiciones dadas. Así si la lista es [10, 20, 5, 4, 6, 8,1] y las posiciones son 2 y 4 la lista resultante sería [5, 4, 6].

5. Escribir una función iterativa llamada `palindromo(lista)` que reciba una lista y retorne True si la lista recibida es un palíndromo y False si no. Una lista es palíndromo si puede leerse igual de izquierda a derecha que de derecha a izquierda. Por ejemplo las listas [3, 8, 5, 8, 3] y [2, 4, 4, 2] son palíndromos.

6. Escriba una función `mitades(lista)` usando iteración que reciba una lista constituida de números enteros y que devuelva verdadero (True) si la suma de la primera mitad de los enteros de la lista es igual a la suma de la segunda mitad y falso en caso contrario. En caso que la longitud de la lista sea impar, NO se considera el elemento de la mitad.

```
>>> mitades([1, 2, 4, 8, 0, 1, 6])
True
>>> mitades([4, 3, 2, 3, 1, 4])
False
```

7. Escriba una función `ultimo_par(lista)` utilizando iteración, que recibe una lista no nula y retorna el último número par de la lista dada. Asuma que la lista está compuesta solo por números.

```
>>> ultimo_par([23, 72, 149, 34])
34
>>> ultimo_par([12, 5, 21, 3])
12
>>> ultimo_par([1, 3, 5, 7])
```

8. Escriba una función iterativa parejas(num) en la cual num es un valor entre 2 y 100 que encuentre todas las parejas (sin importar el orden) de números que suman el número dado. Ejemplos:

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

9. Escriba una función iterativa llamada `digitos_ele(lista)` que reciba una lista de números enteros y retorne una lista que contenga la sumatoria de los dígitos de cada número:

```
>>> digitos_ele([48240, 1, 786])
[18, 1, 3]
```