

PRÁCTICA Nº 4

Cátedra Programación II

Agosto 2017

1. Listas

1. Escriba una función `posicionesMultiplo` que tome una lista y, un número, y retorne la lista formada por los elementos que están en las posiciones múltiplos de ese número. Por ejemplo: `posicionesMultiplo([1,2,3,4,5,6,7],2)` retorna `[1,3,5,7]` y, `posicionesMultiplo([1,2,3,4,5,6,7],3)` da como resultado `[1,4,7]`.
2. Escriba una función que tome una lista de números y devuelva la suma acumulada, es decir, una nueva lista donde el primer elemento es el mismo, el segundo elemento es la suma del primero con el segundo, el tercer elemento es la suma del resultado anterior con el siguiente elemento y así sucesivamente. Por ejemplo, la suma acumulada de `[1,2,3]` es `[1,3,6]`.
3. Escriba una función llamada `elimina` que tome una lista y elimine el primer y último elemento de la lista y cree una nueva lista con los elementos que no fueron eliminados.
4. Escriba una función `ordenada` que tome una lista como parámetro y devuelva `True` si la lista está ordenada en orden ascendente y devuelva `False` en caso contrario. Por ejemplo, `ordenada([1, 2, 3])` retorna `True` y `ordenada(['b', 'a'])` retorna `False`.
5. Escriba una función llamada `duplicado` que tome una lista y devuelva `True` si tiene algún elemento duplicado. La función no debe modificar la lista.
6. Escriba una función llamada `eliminaDuplicados` que tome una lista y devuelva una nueva lista con los elementos únicos de la lista original. No tienen porque estar en el mismo orden.
7. Escriba una función que tome una lista y retorne la cantidad de elementos distintos que tiene. Se recomienda usar la función anterior.
8. Para comprobar si una palabra está en una lista se puede utilizar el operador **in**, pero sería una búsqueda lenta, ya que busca a través de las palabras según el orden que están en la lista. Si la lista almacena las palabras en orden alfabético, podemos acelerar las cosas con una búsqueda dicotómica (también conocida como búsqueda binaria), que es similar a lo que hacés cuando buscas una palabra en el diccionario. Comenzamos por el centro y comprobamos si la palabra que buscamos está antes o después del centro. Si está antes, se busca solo en la primera mitad, si está después se busca en la otra mitad de la lista. Con esto reduciremos el tiempo de búsqueda.

Se pide que implemente la función `busquedaDicotomica` que toma una lista de palabras ordenadas alfabeticamente y, una palabra a buscar y, retorna si la palabra está en la lista o no.

2. Cadenas

- 1) Escriba un programa que tenga una función que tome una cadena y muestre cada caracter que la forma del último al inicial.
- 2) Escriba un programa que contenga a la función `contar(l, x)` que cuente cuántas veces aparece un carácter `l` dado en una cadena `x`.
- 3) Escriba un programa que cuente cuántas veces aparecen cada una de las vocales en una cadena. No importa si la vocal aparece en mayúscula o en minúscula.
- 4) Escriba un programa que contenga la función que reciba como parámetro una cadena de palabras separadas por espacios y devuelva, como resultado, cuántas palabras de más de cinco letras tiene la cadena dada.