

Ejercicios de Python III

Crear el fichero `palabras.py` y añadir el siguiente código:

1. La función `devPalabras()` recibe una cadena con el nombre de un fichero y devuelve una lista con todas las palabras de dicho fichero. Para ello crear previamente un fichero texto (`palabras.txt`) con varias palabras dentro.
2. La función `seleccionaElemento()` devuelve un elemento al azar de una lista de elementos que recibe como parámetro.
3. La función `introLetra()` pide al usuario por teclado una letra, comprueba si efectivamente es una única letra y la devuelve. Si el usuario introduce más de una letra, la función volverá a pedir al usuario que introduzca una letra.
4. La función `buscaLetra()` recibe como parámetros dos cadenas. El primero será una letra y el segundo una cadena. La función devolverá una lista con las posiciones dentro de la cadena donde se encuentre la letra.
5. Hacer finalmente un programa que use las funciones anteriores de la siguiente forma:
 - a) Recoja en una lista (llamada `palabras`) las palabras de un fichero llamando a la función `devPalabras()`.
 - b) Recoja en una cadena (llamada `palabra`) una palabra de la lista anterior llamando a la función `seleccionaElemento()`.
 - c) Recoja una letra (llamada `letra`) del usuario llamando a la función `introLetra()`.
 - d) Crea una cadena (llamada `pista`) con la misma longitud que la palabra seleccionada pero ocultando todas sus letras con el símbolo del guión (-).
6. Con estas funciones es ahora muy sencillo continuar el programa para hacer el juego del ahorcado. Continúa el programa y haz una versión de este juego antes de ver la solución propuesta en el siguiente punto.
7. Solución propuesta. Después de ejecutar a), b), c) y d), ejecutar:

```
while 1:
    letra=introLetra()
    listaLetra=buscaLetra(letra,palabra)
    for i in listaLetra:
        pista=pista[0:i]+letra+pista[i+1:]
    print pista
    if "-" not in pista: break
```

8. Analiza el código anterior y compáralo con el tuyo. De las dos versiones elige el más consistente con Python, el de mayor calidad, el más breve o el que más te guste.
9. Añadir al programa lo necesario para que contabilice el número de fallos y el número de aciertos.
10. Ídem para el tiempo de ejecución total del programa.
11. Modificar el programa para que elimine las palabras repetidas del fichero de entrada.