

Entrega 2: Ejercicios cadenas de caracteres y listas

Ejercicio 1

Realiza un programa que pida una cadena. A continuación debe pedir otra cadena. El programa debe buscar la segunda cadena en la primera (ignorando mayúsculas o minúsculas) y podrá responder una de las siguientes opciones:

- La segunda cadena es una subcadena de la primera
- La segunda cadena no es una subcadena de la primera

Ejemplo:

Cadena 1: Java es un lenguaje de programación

Cadena 2: LENGUAJE

Respuesta:

La segunda cadena es una subcadena de la primera

Ejercicio 2

Escribe un programa que pida una cadena de caracteres y diga si *no* tiene caracteres repetidos.

Ejercicio 3

Tenemos la siguiente variable definida en nuestro programa:

```
temperaturas=""  
    Utrera,29,12  
    Dos Hermanas,32,14  
    Sevilla,30,15  
    Alcalá de Guadaíra,31,14  
""
```

En esa cadena se definen nombres de poblaciones y las temperaturas máximas y mínimas de dichas poblaciones durante un día.

Realiza un programa que muestre el nombre de las poblaciones y la temperatura media. Además el programa te debe pedir el nombre de una población y nos debe dar la temperatura máxima y mínima (si la población no existe se debe dar un error.)

Ayuda: Puede venir muy bien utilizar los métodos splitlines y split de cadenas.

Ejercicio 4

Escriba un programa que permita crear una lista de palabras y que, a continuación de tres opciones:

- Contar: Me pide una cadena, y me dice cuantas veces aparece en la lista
- Modificar: Me pide una cadena, y otra cadena a modificar, y modifica todas a las apariciones de la primera por la segunda en la lista.
- Eliminar: Me pide una cadena, y la elimina de la lista.

El programa te muestra el menú, hasta que introduzcamos la opción 0 de salir.

Dígame cuántas palabras tiene la lista: 4

Dígame la palabra 1: Carmen

Dígame la palabra 2: Alberto

Dígame la palabra 3: Benito

Dígame la palabra 4: Carmen

La lista creada es: ['Carmen', 'Alberto', 'Benito', 'Carmen']

Elige opción:

1. Contar
2. Modificar
3. Eliminar
0. Salir

1

Dígame la palabra a buscar: Carmen

La palabra 'Carmen' aparece 2 veces en la lista.

2

Sustituir la palabra: Carmen

por la palabra: David

La lista es ahora: ['Alberto', 'David', 'Benito', 'David']

3

Palabra a eliminar: David

La lista es ahora: ['Alberto', 'Benito']

0

Adiós!!!

Ejercicio 5

Escriba un programa que permita crear dos listas de palabras y que, a continuación, escriba las siguientes listas (en las que no debe haber repeticiones):

- Lista de palabras que aparecen en las dos listas.
- Lista de palabras que aparecen en la primera lista, pero no en la segunda.
- Lista de palabras que aparecen en la segunda lista, pero no en la primera.
- Lista de palabras que aparecen en ambas listas.

Nota: Para evitar las repeticiones, el programa deberá empezar eliminando los elementos repetidos en cada lista.

Dígame cuántas palabras tiene la primera lista: 4

Dígame la palabra 1: Carmen

Dígame la palabra 2: Alberto

Dígame la palabra 3: Benito

Dígame la palabra 4: Carmen

La primera lista es: ['Carmen', 'Alberto', 'Benito', 'Carmen']

Dígame cuántas palabras tiene la segunda lista: 3

Dígame la palabra 1: Benito

Dígame la palabra 2: Juan

Dígame la palabra 3: Carmen

La segunda lista es: ['Benito', 'Juan', 'Carmen']

Palabras que aparecen en las dos listas: ['Carmen', 'Benito']

Palabras que sólo aparecen en la primera lista: ['Alberto']

Palabras que sólo aparecen en la segunda lista: ['Juan']

Todas las palabras: ['Carmen', 'Benito', 'Alberto', 'Juan']

Ejercicio 6

Realizar un programa que guarde en una lista los nombre y edades de los alumnos de una clase. El programa irá pidiendo por teclado el nombre (string) y la edad (int) hasta que se introduzca como nombre un "*". Las posiciones pares (0,2,4,...) de la lista serán cadenas y las impares son enteros. Cuando terminemos de meter datos hay que mostrar la siguiente información:

- Los nombres de los alumnos con más edad.
- La media de edad de la clase
- Te pide por teclado un nombre y te dice la edad que tiene. Si hay varios alumnos con el mismo nombre te muestra todos.
- Genera una nueva lista con los nombres y edades de los mayores de edad.

Ejercicio 7

Repita el ejercicio 6, pero utilizando la siguiente estructura: una lista, en la cual cada elemento es una lista con dos elementos: el nombre y la edad. Por ejemplo:

```
[ ["juan",18],["maría",21],["pablo",15] ]
```