S2-Tablas-de-Hash

Buscar un elemento en una tabla - Básico

Buscar un elemento en una tabla - Básico

Ficheros requeridos: hasht.py (Descargar)
Tipo de trabajo: Individual

Dada una tabla hash (diccionario de python) y una llave a buscar retornar el valor que se encuentra asociado a dicha llave. Si la llave no existe, retornar None.

Ejemplo:
Entrada:
{"hola": 5,
"pepito": 8,
"jose": -1}
jose
Salida
-1

Ficheros requeridos

hasht.py

```
1 def hasht(table,val):
2     #codigo aqui
```

Función Planta Vieja - Básico

Función Planta Vieja - Básico

Número máximo de ficheros: 1
Tipo de trabajo: Individual

Cree dentro del programa una función que se llame **plantavieja**, y que reciba como parámetro un arreglo de 3 diccionarios.
Se recibe el arreglo de 3 diccionarios como parámetro, NO se recibe por pantalla usando input( ), sino como parámetro, es decir:

```
def plantavieja( elArregloDe3DiccionariosComoParametro : list) -> str :
  ...
```

El arreglo de 3 diccionarios tendrá la siguiente estructura:

[
{"nombre":"Planta1","antiguedad":10},
{"nombre":"Planta2","antiguedad":100},
{"nombre":"Planta3","antiguedad":300},
]

Retorne el nombre de la planta que tiene mayor antigüedad de las 3 plantas recibidas.
Retornar no es imprimir. En inglés, imprimir es *print* y retornar es *return*.

Ejemplo:

[
{"nombre":"Planta1","antiguedad":10},
{"nombre":"Planta2","antiguedad":100},
{"nombre":"Planta3","antiguedad":300},
]

Retorno: *Planta3*

Notas:

1) Elabore su programa en un archivo de Python (llamado **plantas.py**). Si omite la extensión le aparecerá un error de compilación. No deje

Función Chequeo Diccionario - Básico

Función Chequeo Diccionario - Básico

Número máximo de ficheros: 1
Tipo de trabajo: Individual

Cree dentro del programa una función que se llame **chequeodiccionario**, y que reciba un diccionario.
El diccionario tendrá la siguiente estructura

{"nombre":"Nombre maquina","precio":400}

Retorne True si el precio almacenado en el diccionario es de 300 o más, y si el nombre no empieza por la letra C.
De lo contrario, retorne False.

Ejemplo: *{"nombre":"Caladora","precio":400}*

Retorno: *False*

Notas:

1) Elabore su programa en un archivo de Python (llamado **midiccionario.py**). Si omite la extensión le aparecerá un error de compilación. No

***S2* - Números que NO están en dos arreglos - Sencillo**

S2 - Números que NO están en dos arreglos - Sencillo

Ficheros requeridos: arreglo.py (Descargar)
Tipo de trabajo: Individual

Dados dos arreglos de números enteros, retornar un arreglo con los números que NO se encuentran en común en los dos arreglos. Es decir, los que NO están en común. NO, no es los que están en común, sino los que NO están en común.

Entrada
Dos arreglos como parámetro de una función. No, no hay que recibir nada por pantalla.

Salida
Retornar un arreglo ordenado con los números que no aparecen en ambos arreglos. No, no hay que imprimir nada.

Ejemplo:

Entrada:
3 2 1 2 1 4 5 8 6
1 5 7 8 3 9 100 3 4

Salida:
2 6 7 9 100

Ficheros requeridos

arreglo.py

```
1 def arreglo(arr1,arr2):
2     #codigo aqui
```

***S2* - Números que están en dos arreglos - Sencillo**

S2 - Números que están en dos arreglos - Sencillo

Ficheros requeridos: arreglo.py (Descargar)
Tipo de trabajo: Individual

Dados dos arreglos de números enteros, retornar un arreglo con los números que se encuentran en los dos arreglos; es decir, los números que están en ambos arreglos.

Entrada
Dos arreglos como parámetro de una función. No, no hay que recibir nada por pantalla.

Salida
Retorna un arreglo ordenado con los números que se repiten en ambos arreglos. No, no hay que imprimir nada en la pantalla.

Ejemplo:

Entrada:
3 2 1 2 1 4 5 8 6
1 5 7 8 3 9 100 3 4

Salida:
1 3 4 5 8

Ficheros requeridos

arreglo.py

```
1 def arreglo(arr1,arr2):
2     #codigo aqui
```

***S2* - El primer caracter que se repite 2 veces - Sencillo**

S2 - El primer caracter que se repite 2 veces - Sencillo

Ficheros requeridos: caracter.py (Descargar)
Tipo de trabajo: Individual

Dada una cadena, retornar cuál es el primer caracter que se repite dos veces.

Entrada
Una cadena como parámetro de función. No, no hay que recibir nada por pantalla.

Salida
Retorna el primer caracter que se repite dos veces. No, no hay que imprimir nada en la pantalla.

Ejemplo:

Entrada:
sossaa

Salida:
s

Ficheros requeridos

caracter.py

```
1 def caracter(cadena):
2     #codigo aqui
```

***S2* - Extensión de ciudades - Intermedio**

S2 - Extensión de ciudades - Intermedio

Ficheros requeridos: Extension.py (Descargar)
Tipo de trabajo: Individual

Con el objetivo de hacer un análisis territorial en Colombia, se pretende crear un programa que permita identificar la cantidad de ciudades que tienen entre sí una diferencia de área establecida, para ello se cuenta con un arreglo que contiene la extensión territorial (en km^2) de varias ciudades de Colombia. Deberás crear un programa que reciba dicho arreglo (1<=arreglo.length<=1100) y que devuelva cuantos pares de ciudades se llevan una diferencia territorial k (15<= k <= 65200).

Ejemplo:

Entrada

arreglo= {1110, 956, 1780, 1030, 1020, 1120, 878}
k=10

Salida

2

Explicación: Hay dos parejas (1110,1120) y (1030,1020) tales que la diferencia entre ellas es 10.

Nota:

- Recuerde que el ejercicio es totalmente INDIVIDUAL.
- El ejercicio debe solucionarse con TABLAS DE HASH.
- El **array** y el **k** se pasan como parámetros a la función que debes crear.

Ficheros requeridos

Extension.py

```
1 def extension(arreglo,k):
2     # codigo aqui
```

S2 - La nota - Intermedio

S2 - La nota - Intermedio

Ficheros requeridos: nota.py (Descargar)
Tipo de trabajo: Individual

Mauricio es un secuestrador que escribió una nota de rescate, pero ahora le preocupa que lo puedan rastrear a través de su letra. Mauricio ha encontrado una revista y quiere saber si puede recortar palabras enteras de ella y utilizarlas para crear una réplica, imposible de rastrear, de su nota de rescate.

Las palabras de su nota distinguen entre mayúsculas y minúsculas, y debe utilizar sólo palabras enteras disponibles en la revista. Mauricio no puede utilizar subcadenas ni concatenación para crear las palabras que necesita.

Dadas las palabras de la revista y las palabras de la nota de rescate, imprima "Si", si puede replicar su nota de rescate --exacta-- usando palabras enteras de la revista; de lo contrario, imprima "No".

Ejemplo

Revista = "ataque al amanecer"

Nota = "Ataque al amanecer"

La revista tiene todas las palabras correctas, pero hay una diferencia en mayúsculas. La respuesta es "No".

Entrada

La primera línea contiene dos enteros separados por espacios, m y n, que son el número de palabras en la revista y en la nota, respectivamente.

La segunda línea contiene m palabras separadas por espacios, cada palabra de la revista.

La tercera línea contiene n palabras separadas por espacios, cada palabra de la nota.

1 <= m,n <= 30000

Ejemplo 1:

Entrada

9 6

dame uno de los grandes hoy por la noche

dame uno de los grandes hoy

Salida

Si

Ejemplo 2:

Entrada

6 5

dos por tres no es cuatro

dos por dos es cuatro

Salida

No

Nota 1: Tiene que realizar la lectura e impresión por su cuenta. La lectura se hace utilizando entrada (en inglés, *input*).

Nota 2: Tienes que usar tablas hash; conocidas, en Python, como diccionarios.