## Roturas de stock W90261_es

Escribe una función llamada `roturasdestock`. Dicha función recibirá dos parámetros: una lista de tuplas y un diccionario. Cada tupla tendrá dos elementos: el primero será una cadena de caracteres y el segundo, un número entero. De manera similar, las claves del diccionario serán cadenas de caracteres y sus valores serán números enteros. La función deberá devolver un diccionario cuyas claves serán cadenas de caracteres y cuyos valores serán números enteros.

En el contexto de una tienda online, debemos gestionar el stock de cada producto cada vez que se realiza un pedido. El diccionario que se pasa como parámetro contiene el stock (valor) de cada producto de la tienda (clave)

La lista que recibe como entrada la función representa un pedido. Cada tupla de la lista contiene el nombre del producto que se ha pedido (primer elemento) y el número de unidades pedidas (segundo elemento). La lista no contendrá más de una tupla con el mismo nombre de producto.

La función deberá comprobar para qué productos del pedido el número de unidades pedidas es mayor que el stock, es decir, no hay unidades suficientes para completar el pedido. El diccionario devuelto deberá contener como claves los nombres de los productos para los cuales no hay unidades suficientes como valores las unidades que faltan (número de unidades pedidas menos stock). Además, si algún nombre de producto del pedido no está en el diccionario de stock, consideraramos que hay 0 unidades en stock y, por tanto, también habrá que incluirlo en el diccionario devuelto.

Si hay unidades sufientes de todos los productos del pedido, deberá devolverse un diccionario vacío.

Por ejemplo, para la lista de entrada @[("cafetera Delonghi",1), ("granos Robusta",4)]@ y el diccionario {@"cafetera Delonghi": 5 , "granos Robusta": 4, "monodosis Nespresso": 10@}, el programa deberá devolver un diccionario vacío porque hay stock suficiente de todos los productos.

Para la lista de entrada @[("cafetera Delonghi",2), ("granos Robusta",5), ("monodosis Nespresso": 10) ]@ y el diccionario {@"cafetera Delonghi": 5 , "granos Robusta": 4 @}, habrá que devolver el diccionario {@ "granos Robusta": 1, "monodosis Nespresso": 10 @}.

Para que tu función pueda ser evaluada correctamente por el juez en línea, tu código deberá tener la siguiente forma:

```
import sys

def roturasdestock ( lista , stock ):
 ...


l=eval(sys.stdin . readline (). strip ())
d=eval(sys.stdin . readline (). strip ())
o= roturasdestock (l ,d)
print( "+".join( sorted([ k+":"+str(o[k]) for k in o ]) ) )
```

## Entrada

(Si utilizas el fragmento de código definido más arriba, no debes preocuparte por esto) Dos líneas: la primera contendrá la lista escrita en una sola línea como si se tratara de código fuente Python, la segunda contendrá el diccionario del mismo modo.

## Salida

(Si utilizas el fragmento de código definido más arriba, no debes preocuparte por esto) El contenido del diccionario en una sola línea. Cada elemento del diccionario estará separado por el carácter +. Clave y valor estarán separados por dos puntos. Los elementos estarán ordenados alfabéticamente.

## Información del problema

Autoría: Juan Morales García

Generación: 2026-01-25T18:03:00.893Z