Actualizar stock X27883_es

Escribe una función llamada `actualizarstock`. 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 no debe devolver nada: únicamente debe modificar el diccionario que se ha pasado como parámetro.

En el contexto de una tienda online, debemos actualizar 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, simplemente, modificar el diccionario para reducir el stock de cada producto del pedido de acuerdo a las unidades que se han pedido.

Atención: si para algún producto del pedido, el número de unidades pedidas es mayor que el stock, no habrá que realizar ninguna modificación en el diccionario porque asumiremos que ese pedido no se puede realizar debido a que falta algún producto. Del mismo modo, tampoco habrá que modificar el diccionario si algún producto del pedido no se encuentra en el mismo.

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 modificar el diccionario para que quede el siguiente modo: {@"cafetera Delonghi": 4 , "granos Robusta": 0, "monodosis Nespresso": 10@}.

En cambio, para la lista de entrada @[("cafetera Delonghi",2), ("granos Robusta",5)]@ y el diccionario {@"cafetera Delonghi": 5 , "granos Robusta": 4, "monodosis Nespresso": 10@}, el diccionario debe permanecer intacto porque no hay suficiente stock de "granos Robusta". Del mismo modo, para la lista de entrada @[("coche de carreras",1), ("granos Robusta",2)]@ y el diccionario {@"cafetera Delonghi": 5 , "granos Robusta": 4, "monodosis Nespresso": 10@}, el diccionario también permanecerá intacto porque el producto "coche de carreras" no está en el diccionario.

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

```
import sys

def actualizarstock ( lista , stock ):
 ...

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

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: 2026-01-25T18:47:35.196Z