
Intersección de stock**X87176_es**

Escribe una función llamada `intersecciondestock`. Dicha función recibirá dos diccionarios como parámetros. Las claves de los diccionarios 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 en cada uno de los almacenes disponibles. Los diccionarios que se pasan como parámetros representan el stock en dos almacenes distintos. Las claves son los nombres de productos y los valores representan el número de productos disponibles en el almacén. La empresa está pensando en centralizar sus dos almacenes en uno solo, y quiere calcular qué cantidad de cada producto deberá guardar en el nuevo almacén. Además, quiere deshacerse de productos con poca demanda, que son los que sólo se encuentran en uno de los dos almacenes.

La función deberá comprobar qué productos se encuentran en ambos diccionarios de entrada y devolver un nuevo diccionario que contenga como claves únicamente esos productos. Los valores se calcularán como la suma de las unidades disponibles en ambos almacenes.

Por ejemplo, para los diccionarios de entrada `{"cafetera Delonghi": 2, "granos Robusta": 4 @}` y `{"cafetera Delonghi": 5, "granos Robusta": 4, "monodosis Nespresso": 10@}`, el programa deberá devolver el diccionario `{"cafetera Delonghi": 7, "granos Robusta": 8 @}`

Para los diccionarios de entrada `{"cafetera Krups": 2, "granos Arabica": 4 @}` y `{"cafetera Delonghi": 5, "granos Robusta": 4, "monodosis Nespresso": 10@}`, el programa deberá devolver un diccionario vacío porque no hay productos en común.

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

```
import sys
```

```
def intersecciondestock (stock1 ,stock2 ):
```

```
...
```

```
s1=eval(sys.stdin . readline () . strip ())
```

```
s2=eval(sys.stdin . readline () . strip ())
```

```
o= intersecciondestock (s1,s2)
```

```
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 el primer diccionario escrito en una sola línea como si se tratara de código fuente Python, la segunda contendrá el segundo 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: Víctor M. Sánchez Cartagena

Generación: 2026-01-25T19:52:01.256Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>