

---

**Extraer DNIs****X88021\_es**

---

Escribe una función llamada `extraerdnis`. Dicha función recibirá como único parámetro una cadena de caracteres. La función deberá devolver un diccionario cuyas claves serán cadenas de caracteres y cuyos valores serán números enteros.

La cadena de caracteres recibida como parámetro será una oración en castellano, en la cual los signos de puntuación estarán separados de las palabras por espacios en blanco. Por ejemplo: Vivo en el número 29 de la calle Falsa . Mi DNI es 12345678A , y el de CIF de mi empresa es Q-0332001-G . Por si no ha quedado claro , mi DNI es 12345678A . El DNI de mi jefe es 12245678B .

La función deberá identificar y contar en la oración aquellas palabras que son DNIs. Para identificar palabras, simplemente habrá que trocear la cadena utilizando como separador el espacio en blanco. Los DNIs serán palabras compuestas por exactamente 8 dígitos y una letra mayúscula, en este orden. No habrá que comprobar que la letra es la que corresponde a los números según el algoritmo oficial para calcular la letra del DNI.

El diccionario que devolverá la función contendrá, como claves, los DNIs que aparecen en la oración y, como valores, el número de veces que aparece cada DNI. Si la oración de entrada no contiene DNIs, habrá que devolver un diccionario vacío.

Por ejemplo, para la cadena de entrada mencionada anteriormente, la función deberá devolver el diccionario `{@"12345678A": 2, "12245678B": 1@}`. Ten en cuenta que el orden de las claves en un diccionario es irrelevante, así que no te preocupes si, al imprimir el diccionario producido por tu función, obtienes `{@"12245678B": 1, "12345678A": 2@}`.

Si se recibe como entrada una lista vacía, la función deberá devolver un diccionario vacío.

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

```
import sys
```

```
def extraerdnis (c):
```

```
...
```

```
c=sys.stdin.readline().strip()
```

```
resultado=extraerdnis(c)
```

```
print( "+" .join( sorted([ k+" ":" "+str(resultado[k]) for k in resultado ]) ) )
```

**Entrada**

(Si utilizas el fragmento de código definido más arriba, no debes preocuparte por esto) Una línea que contendrá la oración de entrada.

**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: 2026-01-25T19:52:58.534Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>