

---

**El Coleccionista de Letras Únicas****P26915\_es**

---

Un peculiar bibliotecario tiene la costumbre de organizar los libros basándose en secuencias de letras únicas. Su teoría es que las palabras más interesantes son aquellas que contienen la mayor cantidad posible de letras diferentes consecutivas. Para ayudarle, necesitamos crear un programa que analice cadenas de texto y encuentre la subcadena más larga donde no se repita ningún carácter. Por ejemplo, en la cadena “cerebro”, la subcadena más larga con caracteres únicos es “ebro”. Cada letra mayúscula se considera diferente de su versión en minúscula.

**Entrada**

La entrada consiste en varios casos de prueba. Cada caso de prueba contiene una única línea con una cadena no vacía de longitud igual o inferior a 260000, formada únicamente por caracteres del alfabeto inglés (a–z, A–Z).

**Salida**

Para cada caso de prueba, el programa debe imprimir una línea con dos elementos separados por un espacio: la subcadena más larga con caracteres únicos y la posición donde comienza esta subcadena (numerando desde 0). Si existe más de una solución, se debe imprimir la que aparece primero en la cadena original.

**Ejemplo de entrada 1**

```
cerebro
aaabb
abcdefdabcghi
aAbBcC
```

**Ejemplo de salida 1**

```
ebro 3
ab 2
efdabcghi 4
aAbBcC 0
```

**Observación**

Recordad que las funciones `ord` y `chr` permiten convertir entre un carácter y su código numérico ASCII: por ejemplo, `chr(ord('x')+1)` devuelve `'y'`. El mayor código numérico de las letras consideradas es el de la `'z'`, y el menor el de la `'A'`.

**Observación**

Se espera una solución en tiempo lineal, que procese cada carácter rápidamente sin importar el número total de letras distintas en el alfabeto.

**Información del problema**

Autoría: David García Soriano

Generación: 2026-01-25T10:30:21.366Z

© Jutge.org, 2006–2026.

<https://jutge.org>