

Cadenas expansivas



En este problema vamos a considerar las letras mayúsculas del alfabeto inglés (es decir, sin tildes y sin letra Ñ) y vamos a asignar un *código* numérico a cada letra siguiendo el orden del abecedario. Por ejemplo, a la letra *A* le corresponde el código 0, a la letra *B* el código 1, y así sucesivamente hasta llegar a la *Z*, con código 25. Utilizaremos la notación $code(x)$ para hacer referencia al código de la letra x .



Expandir una cadena consiste en insertar, entre cada par x, y de letras adyacentes, la letra cuyo código es $(code(x) + code(y)) \bmod 26$. Por ejemplo, el resultado de expandir la cadena *ACB* es *ACCDB*, donde se han resaltado las letras insertadas. En efecto, hemos insertado una *C* entre las letras *A* y *C* porque $(code(A) + code(C)) \bmod 26 = (0 + 2) \bmod 26 = 2$, que corresponde a la letra *C*. De modo análogo, hemos insertado una *D* (código 3) entre las letras *C* y *B*, ya que $(code(C) + code(B)) \bmod 26 = (2 + 1) \bmod 26 = 3$.

Estamos interesados en conocer cómo evoluciona una cadena cuando la expandimos de manera reiterada. Por ejemplo, vamos a expandir tres veces la cadena *ACB*:

$$ACB \xrightarrow{1} ACCDB \xrightarrow{2} ACCECFDEB \xrightarrow{3} ACCECGEGCHFIDHEFB$$

En la cadena resultante (*ACCECGEGCHFIDHEFB*), la letra que más se repite es la *C*, que aparece cuatro veces.

1. Implementa una función con la siguiente cabecera:

```
pair<char, int> mayor_frecuencia(const string &palabra_inicial, int n);
```

La función debe averiguar cuál es el carácter más frecuente en la palabra que se obtendría tras expandir n veces la *palabra_inicial* pasada como parámetro, y devolver dicho carácter junto con el número de veces que aparece en la palabra resultante. Por ejemplo, la llamada `mayor_frecuencia("ACB", 3)` debe devolver el par `{'C', 4}`. En caso de empate, debe devolverse la letra que aparece antes en el abecedario (es decir, aquella que tiene un código menor).

2. Indica el coste en tiempo de la función anterior.

Entrada

La entrada consta de una serie de casos de prueba, cada uno en una línea. Cada caso de prueba contiene la palabra inicial (de longitud ≥ 2) formada por letras mayúsculas del alfabeto inglés y un número N ($0 \leq N \leq 35$), que indica cuántas veces queremos expandirla. Se garantiza que, en el resultado de la expansión, ninguna letra se repite más de 10^9 veces.

Salida

Para cada caso de prueba debe escribirse una línea con la letra que más se repite en la palabra resultante de aplicar N veces el proceso de expansión sobre la palabra inicial, seguida del número de veces que aparece la letra en dicha palabra.

Entrada de ejemplo

ACB 3
AN 5
CC 4

Salida de ejemplo

C 4
N 22
K 4

Créditos

Manuel Montenegro.