

Clasificando correspondencia

Contribución de Natalia Pérez y Laura Rivero

Descripción del problema

En el correo de una gran ciudad, la clasificación de la correspondencia de acuerdo a su tipo, es un gran problema debido a los volúmenes de cartas, postales, y afines que manejan.

Afortunadamente cuentan con un sistema que coloca los sobres y paquetes en una cinta sinfín, que luego pasan por un escáner que permite definir su tipo (carta, sobre oficio, tarjeta postal, caja pequeña, etc.). A los efectos de procesar las cantidades de correspondencia que reciben, a cada tipo le asignan una letra minúscula. De esta manera, los objetos que van pasando por el escáner generan una secuencia de letras que el sistema luego tiene que interpretar.

Los responsables del manejo de la correspondencia desean guardar diariamente, información relativa al tipo y cantidad de objetos recibidos. De esta manera han ideado una codificación que transforma las repeticiones consecutivas de letras (que representan un tipo), en un par letra y número (que indica la cantidad de repeticiones). Por ejemplo, si la secuencia de letras generada por el escáner es **aaacyyyyyyybbuccaayy**, se codificaría como **a3c1y8b2u1c2a2y2**. Para hacer el reporte diario se requiere obtener, además de la secuencia original codificada, la lista de cantidades totales por tipo, ordenada alfabéticamente.

Para ayudar a los empleados del correo se te pide que escribas un programa **sinfin.c**, **sinfin.cpp** o **sinfin.pas** que, a partir de la secuencia de letras generada por el escáner, obtenga la codificación de la misma y la cantidad total de correspondencia de cada tipo, ordenada alfabéticamente.

Datos de entrada

Se recibe un archivo **sinfin.in** con:

- Una línea indicando la cantidad **L** ($1 \leq L \leq 200$), de letras que contiene la secuencia generada por el escáner.
- 1 línea con la secuencia de letras minúsculas.

Datos de salida

Se debe generar un archivo **sinfin.out** que contendrá

- Una línea con la secuencia codificada.
- Tantas líneas como sean necesarias, cada una de ellas con el par letra-cantidad total de apariciones separados por blanco. Los pares deben aparecer ordenados alfabéticamente.

Puntuación

Una respuesta correcta recibe **100** puntos. Se asigna un **50%** del puntaje a la primera línea, y otro **50%** a la lista.

Ejemplo

Si la entrada **sinfin.in** fuera:

```
21
aaacyyyyyyybbuccaayy
```

La salida **sinfin.out** sería:

```
a3c1y8b2u1c2a2y2
a 5
b 2
c 3
u 1
y 10
```