

---

**Examen práctica - Turno 1 - Problema 2 - Actualizació de tablas de frecuencias**

---

**X30943\_es**

---

1. El peso de este ejercicio en la nota del examen de la práctica es de un 66.66% (2/3 de la nota).
2. Nota manual: 50%, nota automática: 50%
3. El peso de todos los juegos de pruebas en el cálculo de la nota automática es idéntico (público: 2.5/10, privados: 2.5/10 cada uno).

Dadas dos tablas de frecuencias  $f1 = F_1$  y  $f2 = F_2$ , queremos una función que actualiza la primera con los datos de la segunda. Las tablas estarán representadas por `map<string, int>`. Todas las frecuencias en  $f1$  y  $f2$  son estrictamente positivas.

Como resultado de la actualización la tabla  $f1$  contendrá:

- Si  $\langle s, f \rangle$  es un par de la tabla  $F_1$  y  $F_2$  no contiene ningún par con clave igual a  $s$ , entonces  $\langle s, f \rangle$  está en la tabla  $f1$  final.
- Si  $\langle s, f \rangle$  es un par de la tabla  $F_2$  y  $F_1$  no contiene ningún par con clave igual a  $s$ , entonces  $\langle s, f \rangle$  está en la tabla  $f1$  final.
- Si  $\langle s, f \rangle$  es un par de la tabla  $F_1$  y  $\langle s, f' \rangle$  es un par de la tabla  $F_2$  entonces  $\langle s, f + f' \rangle$  está en la tabla  $f1$  final.

Por ejemplo si  $f1$  y  $f2$  son las tablas mostradas a continuación:

a	11
b	3
d	7

b	6
c	17
d	2
e	2

entonces después de la actualización  $f1$  será

a	11
b	9
c	17
d	9
e	2

Implementa la siguiente función

```
// Pre: y Post: ver la descripción del enunciado
void actualiza_tabla_frec(map<string,int>& f1, const map<string,int>& f2);
```

Escribe un pequeño programa que lea una tabla de frecuencias  $f1$ , y a continuación, aplica sucesivas actualizaciones, imprimiendo cómo queda  $f1$  después de cada actualización. Para cada actualización se ha de leer una tabla de frecuencias  $f2$ , actualizar  $f1$  con  $f2$  mediante la función implementada, e imprimir la tabla  $f1$ .

N.B. De cara a obtener una solución lo más eficiente posible, tened en cuenta lo siguiente:

1. Recorrer un `map` mediante iteradores de principio a fin toma tiempo proporcional al tamaño del `map`—suponiendo que en cada iteración del recorrido se invierte un tiempo constante.
2. La clase `map` tiene un método `insert(iterator, par-clave-valor)` para dar una “pista” sobre dónde debe insertarse el nuevo elemento. Por ejemplo, si insertamos  $N$  elementos en orden creciente de clave usando `m.insert(m.end(), x)` el coste será proporcional a  $N$ ; si se utilizase el método normal `m.insert(x)` el coste será mucho mayor, proporcional a  $N \cdot \log_2 N$ .

## Entrada

Cada tabla de frecuencias se representará en la entrada mediante una secuencia que comienza con un entero  $k \geq 0$  seguida de una secuencia de  $k$  pares  $\langle s, f \rangle$  describiendo los pares que constituyen la tabla.

La entrada comienza con la subsecuencia que representa a la tabla  $f_1$ . A continuación viene una serie de tablas de frecuencias (cada una representada por una secuencia con el formato que hemos descrito arriba). La última tabla de frecuencias de la serie es una tabla vacía ( $k = 0$ ).

## Salida

Para cada tabla de frecuencias  $f_2$  de la serie, excepto la última tabla vacía que marca el final de la serie, se actualiza la tabla  $f_1$  con los datos de  $f_2$  y se imprime la tabla  $f_1$  actualizada. Una tabla de frecuencias se imprime siguiendo el mismo convenio que para la entrada: primero un entero  $k$  con su tamaño y a continuación una secuencia de los  $k$  pares  $\langle s, f \rangle$  que contiene la tabla (separamos las componentes  $s$  y  $f$  de cada par mediante un espacio en blanco), pero adicionalmente los pares han de estar en orden creciente de clave (ordenados por la  $s$ ). Después de imprimir la tabla  $f_1$  tras cada actualización se imprime un salto de línea.

## Observación

Utilizad la plantilla (`plantilla-solucion.cc.txt`) que os damos con los ficheros públicos (icono del gatito) para preparar la solución.

## Ejemplo de entrada 1

```
5 a 1 b 1 c 1 d 1 e 1
3 a 1 c 1 f 1
2 b 2 d 2
5 g 1 h 1 a 1 e 1 j 5
0
```

## Ejemplo de salida 1

```
6 a 2 b 1 c 2 d 1 e 1 f 1
6 a 2 b 3 c 2 d 3 e 1 f 1
9 a 3 b 3 c 2 d 3 e 2 f 1 g 1 h 1 j 5
```

## Información del problema

Autoría: Profesores de PRO2

Generación: 2026-01-25T15:04:34.234Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>