

Evaluación

El objetivo de este control es familiarizarse con el manejo de diccionarios.

1) El problema

La evaluación continua se le ha ido de las manos al profesor. Les pide a los alumnos que no lo dejen todo para el final sino que vayan estudiando día a día, pero él no predica con el ejemplo. Ahora tiene todos los ejercicios que los alumnos han ido entregando durante todo el año en una pila de folios y le toca revisarlos. Los ejercicios o están bien (y entonces puntúan positivamente) o están mal (y entonces restan). Al final del día quiere tener imprimida una lista con los nombres de todos los alumnos ordenados en orden alfabético inverso y su puntuación en la evaluación continua (resultado de sumar todos los ejercicios que tienen bien menos los que tienen mal). Si un alumno tiene un 0 como balance no debería aparecer en la lista. Aunque sea abusar un poco del alumnado... ¿puedes ayudar al profesor?

Lo que se pide es que implementes una función que reciba dos listas de cadenas con los nombres de los alumnos. La primera lista tiene un elemento por cada ejercicio correcto entregado y el elemento es el nombre del alumno que lo entregó (por lo tanto, un mismo alumno puede aparecer varias veces, si entregó varios ejercicios bien). De forma similar, la segunda lista contiene los ejercicios incorrectos. La función debe generar una lista de calificaciones a publicar. Dicha lista debe contener el resultado final de la evaluación de los alumnos **cuyo balance es distinto de 0 por orden alfabético inverso**.

2) Trabajo a realizar

Se debe construir un programa que lea una serie de filas, cada una representando los resultados de una serie de alumnos en los ejercicios (las calificaciones se delimitan con #, y todos los elementos en la línea deben ir separados entre sí por un espacio en blanco), e imprima, por cada fila, una línea con el resultado final de la evaluación de los alumnos. La nota de cada alumno se imprimirá con el formato `[nombre_del_alumno:calificación]`, en orden alfabético inverso por el nombre del alumno.

Ejemplo de entrada / salida:

Entrada	Salida
# jose + jose - jose - lucas + pedro - lucas - #	[pedro:-1][jose:-1]
# jose + jose - jose - jose + #	
# #	
# jose - juan + juan + #	[juan:2][jose:-1]

Se proporciona el archivo `main.cpp` en el que se implementa la lógica de entrada/salida necesaria. El código proporcionado no debe modificarse.

Hay que añadir a dicho archivo la implementación de la siguiente función:

```
// Devuelve una lista con las puntuaciones de aquellos alumnos cuya
// calificación final es distinta de 0. Dicha lista está ordenada en orden
// alfabético inverso de nombre de alumno
// Parámetros:
//   bien: Lista de alumnos que han resuelto correctamente un ejercicio (si
//         un alumno ha resuelto correctamente varios ejercicios, aparecerá
//         varias veces en la lista)
//   mal: Lista de alumnos que han resuelto incorrectamente un ejercicio (si
//        un alumno ha resuelto correctamente varios ejercicios, aparecerá
//        varias veces en la lista)
//   listado: Lista de calificaciones, ordenada en orden alfabético inverso
//            de nombre de alumno.
//            Solo aparecen aquellos alumnos cuya calificación final
//            es distinta de 0.
//            La clase Puntuacion permite representar la calificación de
//            un alumno, y está definida en main.cpp.
void califica(const Lista<string>& bien, const Lista<string>& mal,
             Lista<Puntuacion>& listado)
```

Aparte de esta función, podrán añadirse todas aquellas funciones auxiliares que se consideren necesarias.