

Fundamentos de Programación. Doble Grado

Convocatoria de Septiembre. Curso 2012/2013

Tiempo: 2.5 horas

1. (2 puntos) Defina la clase **VectorEnteros**. Basta incluir los datos miembro necesarios y los métodos adicionales que necesite para resolver los siguientes apartados:
 - Defina dentro de la clase lo que sea necesario para sumar dos vectores **-VectorEnteros-** (ambos han de tener el mismo número de componentes). La suma de dos vectores se define como el vector resultante de sumar las correspondientes componentes. Por ejemplo, la suma de $\{2,5,7\}$ y $\{4,2,1\}$ es el vector $\{6,7,8\}$.
 - Defina dentro de la clase lo que sea necesario para calcular el número de secuencias ascendientes del vector. Por ejemplo, el vector $\{2,4,1,1,7,2,1\}$ tiene 4 secuencias que son $\{2,4\}$, $\{1,1,7\}$, $\{2\}$, $\{1\}$.
2. (3 puntos) Se quiere almacenar el nombre de un alumno junto con las notas que ha sacado en varias asignaturas. El número de asignaturas puede variar de un alumno a otro y las notas son de tipo de dato entero. Con estos datos, se quiere construir una clase, es decir, un conjunto de alumnos y ordenarlos de menor a mayor según la media aritmética de sus calificaciones. Por ejemplo, si tenemos

$[Ana \text{ de Gober}, (9,7,8,9)]$, $[Sergio \text{ García}, (3,4,2)]$, $[David \text{ Rodriguez}, (5,4)]$

La nota media de Ana de Gober sería 8.25, la de Sergio García 3 y la de David Rodriguez 4.5 , por lo que el vector resultante debería quedarse como sigue:

$[Sergio \text{ García}, (3,4,2)]$, $[David \text{ Rodriguez}, (5,4)]$, $[Ana \text{ de Gober}, (9,7,8,9)]$

Defina las clases necesarias para resolver este problema. Si lo necesita, puede usar y/o ampliar la clase definida en el primer ejercicio de este examen. No hace falta crear el programa principal ni ninguna clase para la lectura o escritura de los datos.
3. (3 puntos) Defina una clase **Frase** para almacenar un conjunto de caracteres. Como dato miembro ha de usarse un **vector** (de la STL) de **char**. Defina un método para localizar la k -ésima palabra.
 - Una palabra es toda secuencia de caracteres delimitada por espacios en blanco a izquierda y derecha. La primera palabra no tiene por qué tener espacios a su izquierda y la última no tiene por qué tener espacios a su derecha. Puede haber varios caracteres en blanco consecutivos.
 - Si k es mayor que el número de palabras, se considera que no existe tal palabra.
 - Sólo debe definir los datos miembro y métodos necesarios para resolver este ejercicio. No puede usarse el tipo **string**.
 - Por ejemplo, si la frase es $\{ ' ', ' ', 'h', 'i', ' ', ' ', 'b', 'i', ' ', ' ' \}$. Si $k = 1$, la posición es 2. Si $k = 2$ la posición es 6. Si $k = 3$ la posición es -1.
 - Si la frase fuese $\{ 'h', 'i', ' ', 'b', 'i', ' ', ' ' \}$, entonces si $k = 1$, la posición es 0. Si $k = 2$ la posición es 3. Si $k = 3$ la posición es -1.
4. (2 puntos) Implemente una función **RECURSIVA** que acepte un entero y devuelva el vector de enteros menor o iguales que él (hasta llegar al 0). Si se introduce un valor negativo tendrá que devolver un vector formado por dicho negativo. Por ejemplo, si el entero es 4, la función debe devolver el vector $\{0,1,2,3,4\}$. Si el entero es -4, la función debe devolver el vector $\{-4\}$. No puede usarse ninguna estructura repetitiva (bucles **while** o **for**) y se recomienda usar un **vector** de la STL.