

LENGUAJE DE PROGRAMACIÓN 1
7ma práctica (tipo b)
(Segundo Semestre 2018)

Indicaciones Generales:

- Duración: 110 minutos.
- Se podrá usar como material de consulta solo sus apuntes de clase.
- No se pueden emplear variables globales, ni estructuras. No se podrá emplear la clase string ni las funciones malloc, realloc, strdup, strtok, tampoco las funciones incluidas en las bibliotecas cstdio, stdio.h o similares.
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. Cada módulo NO debe sobrepasar las 20 líneas de código aproximadamente. El archivo main.cpp solo podrá contener la función main de cada proyecto. En cada archivo que implemente en los proyectos (.h y .cpp) deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le descontará 0.5 puntos por archivo. Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% del puntaje de la pregunta. Los que no den resultados coherentes en base al 60%.
- La presentación, la ortografía y la gramática de los trabajos influirá en la calificación.

Puntaje total: 20 puntos

Problema

Se tiene un archivo CSV con la siguiente información de una lista de novelas:

| Archivo: "novelas.csv" |
|--|
| Cien años de soledad;Gabriel García Márquez;-0.272209;0.237176;-0.48612... |
| La ciudad y los perros;Mario Vargas Llosa;-0.186415;0.081194;-0.448023... |
| ... |

En cada línea aparece el título de una novela, el nombre de su autor, y 300 valores reales, los que corresponden a un vector que representa dicha novela en un espacio vectorial de 300 dimensiones. Los valores de estos vectores han sido generados con algoritmos de aprendizaje automático, y tratan de mantener una información numérica respecto al texto contenido en las novelas, por lo que serán útiles en una función de comparación que se plantea más adelante.

Pregunta 1 (12 puntos)

Se pide que elabore una aplicación orientada a objetos, denominada Pregunta01, el cual permitirá definir y trabajar con la información de una novela. Por ello, debe definir una clase denominada ClaseNovela, la que se describe a continuación:

```
class ClaseNovela {  
private:  
    char *título;  
    char *autor;  
    double *vector;  
public:  
    ...  
}
```

Cin CC
NO PUEDE AGREGAR
ATRIBUTOS A LA
CLASE NI
MODIFICARLOS.

Esta clase debe incluir la siguiente funcionalidad:

- **Constructor por defecto y destructor.**
- **Métodos selectores.**

Siendo los tres atributos punteros, deberá cuidar que en ningún momento se viole el encapsulamiento de la clase, entregando las direcciones a las que apuntan los atributos o asignando direcciones de memoria a los atributos que son recibidas del exterior. Así como también se deberá verificar que los espacios de memoria que ya no se requieran sean liberados.

En caso del atributo **vector**, como no tiene sentido crear métodos selectores para un arreglo como este, es decir un método al que se le entregue un arreglo para llenar el atributo o un método que devuelva un arreglo con los datos de **vector**, los métodos selectores para este atributo estarán referidos a una celda en particular del arreglo. Es decir, uno de ellos recibirá un valor de punto flotante y un valor entero. El método asignará el valor real en la celda del arreglo **vector** que está en el índice correspondiente al valor entero. El otro método realizará la operación inversa, esto es, se le entrega un valor entero (índice del arreglo) y devuelve el valor de la celda. Puede asumir que el usuario siempre ingresará un índice válido para este problema (entre 0 y 299).

- **Sobrecarga del operador *:**

Este método permitirá determinar la "*similitud*" entre dos libros, siendo este concepto definido como:

$$\text{similitud} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Donde **A** y **B** son dos vectores a comparar, y '*n*' es igual a 300 según este caso.

- **Sobrecarga del operador de asignación (=):**

Debe permitir asignar un objeto de clase **ClaseNovela** a otro.

- **Sobrecarga de los operadores de inserción y extracción de flujo (<< y >>):**

Estas sobrecargas deberán permitir el ingreso y salida de datos desde archivos de texto así como desde la entrada y salida estándar. Considere el formato del archivo CSV para la lectura No podrá emplear la cláusula **friend**.

La función main deberá probar el buen funcionamiento de toda la clase usando el archivo "novelas.csv".

Pregunta 2 (8 puntos)

Se pide que elabore una aplicación orientada a objetos, denominada Pregunta02. Deberá copiar en ella los archivos creados en la pregunta 1 (**ClaseNovela.h** y **ClaseNovela.cpp**). Esta aplicación debe definir una clase denominada **ClaseRecomendacion**, la que se describe a continuación:

```
class ClaseRecomendacion {
private:
    class ClaseNovela *novelas;
    int tam;
    int numNov;
    ...
public:
    ...
}
```

Tam define el tamaño del arreglo novelas
NumNov define el número de novelas en el arreglo.

NO PUEDE AGREGAR ATRIBUTOS A LA CLASE NI MODIFICARLOS.

Esta clase debe incluir la siguiente funcionalidad:

- **Constructor por defecto, constructor con parámetros y destructor.**

El constructor por defecto deberá definir un espacio de memoria fijo para el arreglo. El constructor con parámetros permitirá definir el tamaño del arreglo al usuario.

- **Métodos selectores**

Al igual que la clase anterior, no se definirán métodos para trabajar con todo el arreglo sino con una celda en particular. No se está pidiendo sobrecargar el operador de acceso "[]".

- **Métodos para leer y escribir los datos de las novelas.**

Estos métodos deberán poder leer tanto de la entrada estándar como de un archivo de textos una serie de libros y asignarlos a los elementos del arreglo, así como imprimir los datos de los libros tanto en la salida estándar y como en un archivo de textos. Estos últimos deberán mostrar los datos correctamente tabulados y con etiquetas que muestren la naturaleza de los datos (no podrá usar el carácter '\t').

- **Métodos para ordenar los datos del arreglo.**

Deberá poder ordenar las novelas, por el autor y por el título. Podrá emplear la función `qsort` definida en `cstdlib` o implementar su propio algoritmo pero este deberá tener una eficiencia de $n \log n$.

- **Métodos para consultar las novelas de un autor.**

Deberá poder ingresar un autor y mostrar por la salida estándar de datos todos títulos escritos por el autor.

- **Métodos para consultar los novelas similares**

Deberá poder ingresar el título de una novela y mostrar por la salida estándar los datos de las tres novelas con mayor similitud.

La función `main` deberá probar el buen funcionamiento de toda la clase.

CONSIDERACIONES:

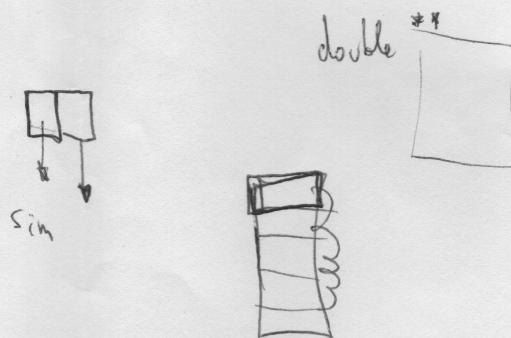
- Cada clase deberá definirse en dos módulos independientes (archivo `.h` y `.cpp`). No se podrá definir la implementación de los métodos en los archivos `.h`.
- Deberá respetar estrictamente el encapsulamiento de datos a todo nivel.
- Con excepción de la sobrecarga de los operadores «`<<`» y «`>>`», NO se podrán definir funciones (ni plantillas) independientes que no estén ligadas como métodos a alguna de las clases planteadas.
- En ambas preguntas, usted podrá definir los métodos y sobrecargas adicionales que crea convenientes.

Anotaciones finales

Al finalizar el laboratorio, comprima¹ la carpeta "Laboratorio7" en un archivo con nombre < código del alumno con 7 dígitos>.zip y súbalo a la intranet del curso, en el enlace Documentos, en la carpeta \Laboratorio7\< código del horario>\aula.

Profesores del curso: Arturo Oncevay
Miguel Guanira E.

San Miguel, 9 de noviembre del 2018.



¹ Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en el Windows (Zip) no 7z.