

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

1er. examen

(Segundo Semestre 2014)

Indicaciones Generales:

Duración: 3 horas.

La presentación, ortografía y gramática influirán en la calificación

Puntaje total: 20 puntos

---

PARTE TEÓRICA

Indicaciones:

- Duración: 15 minutos.
- No se podrá usar material de consulta.
- Puntaje total: 4 puntos

Cuestionario:

Según el texto "C++ How to Program, 7th Edition Chapter ..." de Deitel, responda lo siguiente:

Pregunta 1 (2 puntos)

Defina el concepto de "Interface" y de un ejemplo diferente al mostrado en el libro. Explique por qué es importante separarlo de la denominada "Implementation".

Pregunta 2 (2 puntos)

Defina el concepto de "UML Class Diagram", de un ejemplo diferente al mostrado en el libro en el que se muestre una clase completa explicando cada uno de sus diferentes elementos

---



## PARTE PRÁCTICA

### Indicaciones:

- Duración: 2 horas y 45 minutos.
- Se podrá usar como material de consulta sólo apuntes de clases (no fotocopias ni hojas sueltas).
- No se pueden emplear variables globales, tipos de datos struct, funciones de manipulación de archivos, clases ni objetos, como la clase string. Tampoco se podrán emplear las funciones malloc, realloc, strdup, sscanf, sprintf ni fopen.
- Puntaje total: 16 puntos

Una tienda de libros cuenta con un archivo de texto que contiene una lista de diferentes autores y sus libros publicados. El archivo es similar al siguiente:

436428	Federico	Sánchez	Lozano	
123456	Ana	Cecilia	Roncal	Neyra
...	...	...	...	...
556677	Pedro	Castro	López	de la Romaña
...	...	...	...	...
GR1234PK	Historia	de los últimos	conquistadores	136.75
PT3456MG	Cuentos	de	terror	89.90
VG6677NG	La	pequeña	valiente	145.05
...	...	...	...	...

*Handwritten notes:* "PU abf" and "abi cod abf" with arrows pointing to the price and author code columns respectively. A box highlights the code "123456" in the first column of the last row.

El archivo está dividido en dos partes. La primera parte registra los autores brindando, en cada línea, el código del autor y su nombre. Los códigos son valores enteros y los nombres están separados por uno o más espacios en blanco. No está registrado el número de autores ni hay marcas que indiquen el fin de esta parte.

La segunda parte registra los libros brindando, en cada línea, el código del libro, su título, su precio unitario y el código de su autor. El código del libro es alfanumérico y siempre comenzará con una letra. Un libro sólo registra un autor; sin embargo, un autor puede haber escrito varios libros. Esta parte no tiene orden alguno y tampoco hay marcas que indiquen el final.

Se requiere una aplicación para manejar este archivo, por lo que se le pide a usted que escriba la implementación de las funciones dadas en el siguiente programa:

```
#include <lecturaYPrueba.h>
#include <reporte.h>
int main(void) {
    void *autores,* libros;

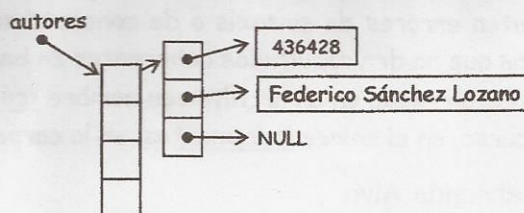
    leerAutores (autores);
    leerLibrosYCompletarAutores (libros, autores);
    reporteFinal (autores, libros);
    return 0;
}
```

En los archivos `lecturaYPrueba` (.h y .cpp) se encontrarán las funciones `leerAutores` y `leerLibrosYCompletarAutores` y las que se requieran para su implementación. Mientras tanto, en los archivos `reporte` (.h y .cpp) se encontrará la función `reporteFinal` y las que se requieran para su implementación.

### Pregunta 3 (3.5 puntos)

#### Función `leerAutores`

Esta función deberá leer, re-direccionando la entrada estándar de datos, la primera parte del archivo descrito anteriormente y guardar la información en la estructura que se muestra a continuación:



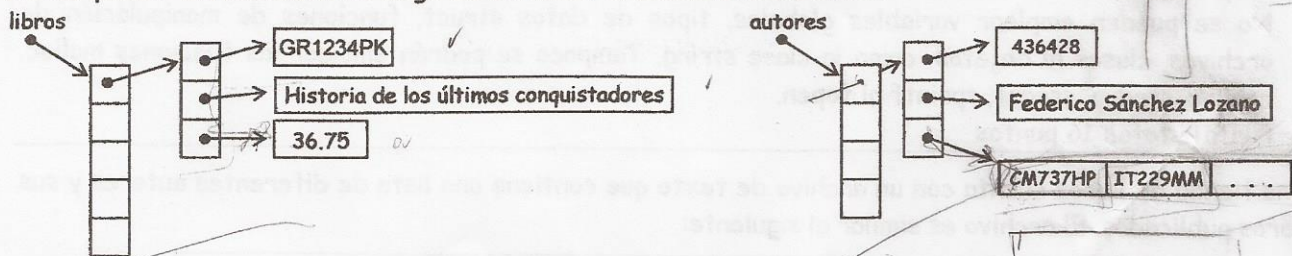
La estructura deberá guardar los datos de manera exacta, eliminando los espacios sobrantes. No puede agregar o modificar los argumentos de la función. Luego de llenar los datos, la función deberá llamar a una función que imprima, de manera clara, todos los datos desde el puntero `autores`.



#### Pregunta 4 (7.5 puntos)

##### Función leerLibrosYCompletarAutores

La función debe leer la segunda parte del archivo, crear la estructura que se muestra a continuación y completar la estructura de autores creada en la pregunta anterior. A continuación se muestra cómo deben quedar las estructuras luego de terminar la lectura de los datos:



La estructura `libros` deberá guardar los datos de manera exacta, eliminando los espacios sobrantes. Luego de llenar los datos, la función deberá llamar a una función que imprima, de manera clara, todos los datos desde el puntero `libros`.

En la estructura `autores`, deberá guardar los códigos de los libros que escribió cada autor. En esta parte los arreglos que contengan los códigos de los libros deberán incrementarse de 5 en 5 (no se asignará puntaje en esta parte si aquí se usan *buffers*). Luego de llenar los datos, la función deberá llamar a una función que imprima, de manera clara, todos los datos desde el puntero `autores`. No puede agregar o modificar los argumentos de la función.

##### Puntaje en esta pregunta:

- Lectura, cargado e impresión de los libros 3.5 puntos.
- Modificación e impresión de los autores 4.0 puntos

#### Pregunta 5 (5 puntos)

##### Función `reporteFinal`

Esta función debe imprimir en un archivo de texto un catálogo de los autores y sus libros contenidos en ambas estructuras. El reporte se deberá ver como en la figura siguiente:

##### Catálogo de autores:

1) 123456 Ana Cecilia Roncal Neyra

##### Libros:

Código	Título	Precio
GR1234PK	Historia de los últimos conquistadores	36.75
VG6677NG	La pequeña valiente	145.05

2) 225566 ...

En el reporte, los autores deberán aparecer ordenados por el código del autor y los libros deberán aparecer ordenados por su código. La ordenación deberá ser hecha empleando Quick Sort o algún otro algoritmo con similar eficiencia ( $n \log n$ ).

#### CONSIDERACIONES FINALES

- LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE, por lo que NO SE CALIFICARÁN aquellas funciones que son llamadas por otras que estén incompletas. Cada módulo no debe sobrepasar las 20 líneas aproximadamente.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% del puntaje de la pregunta y los que no den resultados coherentes en base al 60%.
- Al final, comprima<sup>1</sup> el proyecto creado en un archivo con nombre <código del alumno con 8 dígitos>.zip y súbalo a la intranet del curso, en el enlace Documentos, en la carpeta \Examen1\<aula>.

Profesores del curso: Fernando Alva  
Miguel Guanira

San Miguel, 14 de octubre del 2014

<sup>1</sup> Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en el Windows (Zip).