

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 1

4ta práctica (tipo b)
Segundo Semestre 2015

Indicaciones Generales:

- Duración: 110 minutos.
- Se podrá usar como material de consulta solo sus apuntes de clase.
- No se pueden emplear variables globales, clases, objetos (excepto cin y cout), ni la función fopen.
- Las funciones implementadas en su proyecto deberán estar desarrolladas obligatoriamente en archivo diferentes al main.cpp, debiendo crear los archivos .h y .cpp correspondientes.
- 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.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de 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 con los cursos de una institución, así como una relación con la matrícula de los alumnos de una institución educativa, el archivo es similar al siguiente:

INF281	Lenguajes de programación 1			
MEC386	Mecánica de fluidos 2			
CIV402	Concreto armado 1			
...				
0				
20141005	INF386	INF409	MAT286	IND333...
20090044	CIV212	CON547...		
20101234				
...				

El archivo está dividido en dos partes. La primera parte registra los cursos de la institución, en cada línea aparece el código del curso y su nombre. Al final de esta parte se encontrará una línea con un cero. La segunda parte contiene un listado de matrícula, en cada línea aparece el código de un alumno y las claves de los cursos en los que se ha matriculado. Un alumno puede llevar varios cursos, no necesariamente los alumnos llevan la misma cantidad de cursos.

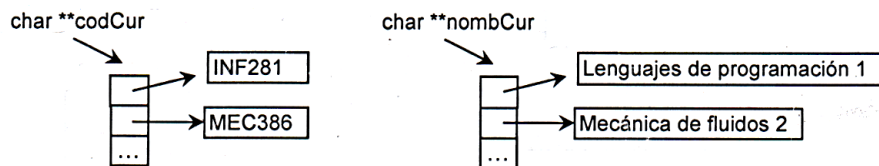
Se quiere una aplicación para manejar este archivo, por lo que se le pide a usted que escriba la implementación de las siguientes funciones dadas en el siguiente programa:

```
/** *****  
/** Bibliotecas de funciones requeridas  
/** *****  
int main()void) {  
    char **codCur, **nombCur;  
    int **listaDeClase, numCursos;  
  
    leerCursos(codCur, nombCur, numCursos);  
    leeMatricula(codCur, numCursos, listaDeClase);  
    listasDeClase(codCur, nombCur, numCursos, listaDeClase);  
    return 0;  
}
```

No puede modificar este código, salvo por las bibliotecas que usará

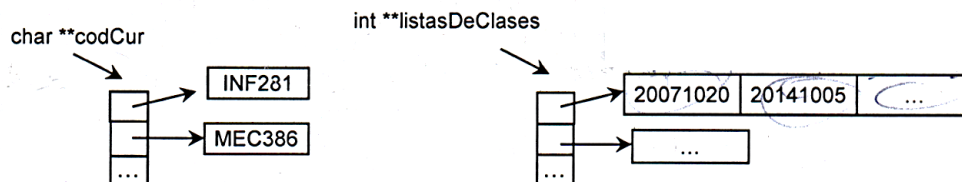
Función leerCursos: (6 puntos)

La función leerCursos deberá leer, re direccionando la entrada estándar de datos, la primera parte del archivo descrito anteriormente y guardar la información en las estructuras que se muestran en la página siguiente, la memoria asignada en todos los casos deberá ser exacta. Antes de retornar, la función deberá imprimir los datos desde los arreglos asignados de manera clara y ordenada.



Función leeMatricula: (9 puntos)

La función `leeMatricula` deberá leer la segunda parte del archivo descrito anteriormente y guardar la información en la estructura que se muestra a continuación:



Cada elemento de la estructura `listasDeClases` corresponderá a un curso y su posición coincidirá con los datos del curso en las otras estructuras. El elemento apuntará a una lista que contendrá los códigos de los alumnos matriculados en ese curso.

Para la asignación de memoria para las listas del arreglo `listaDeClase` deberá emplear el método de incrementos, considerando un incremento de 5 elementos por vez. Como no se devuelve el tamaño ni la capacidad de cada lista, deberá buscar una forma de controlar estos tamaños dentro y fuera de esta función.

Deberá tomar en cuenta que un alumno podrá aparecer varias veces en la estructura.

Antes de retornar, la función deberá imprimir los datos asignados a los arreglos de manera clara y ordenada.

Función listasDeClase: (5 puntos)

La función `imprimeListasDeClase` deberá generar el siguiente reporte:

Listas de clases:		
Clave: INF281	Curso: Lenguajes de programación 1	Número de alumnos: 29
Alumnos:		
1.- 20071020	2.- 20141005	
3.- ...	4.- ...	
...		
Clave: MEC386	...	
...		
...		

Al finalizar el laboratorio, comprima¹ su proyecto 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 que corresponda a su aula.

Profesores: Miguel Guanira E.

San Miguel, 25 de septiembre del 2015.

¹ Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en el Windows (Zip). No use 7-zip o similares.