

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 1

**3ra práctica (tipo b)**

**Segundo Semestre 2013**

**Indicaciones Generales:**

No se pueden emplear variables globales, tipos de datos struct, funciones de manipulación de archivos, clases ni objetos, como la clase string y las funciones que se encuentran en bibliotecas como stringstream. Tampoco se podrán emplear las funciones strtok, strdup, sscanf, sprintf ni fopen.

**NO SE PODRÁ EMPLEAR DISPOSITIVOS USB, NI EL USO DEL CORREO ELECTRÓNICO EN EL LABORATORIO, EL HACERLO IMPLICARÁ LA ANULACIÓN DEL LABORATORIO.**

**Duración:** 1 hora 50 minutos.

**Puntaje total:** 20 puntos.

**Objetivo del laboratorio**

Reforzar en el alumno los conceptos que permiten el manejo de punteros y punteros múltiples, así como el manejo dinámico de datos.

**Caso Práctico:**

Cree una carpeta con nombre *Lab3* y dentro coloque cada una de las partes de este laboratorio.

Se tiene un registro 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  5.5  Lenguajes de programación 1
MEC386  3.0  Mecánica de fluidos 2
CIV402  4.0  Concreto armado 1
...
1 Roncal/Neyra/Ana-Cecilia  INF386  INF409  MAT286  IND333
2 Salazar/Pastor/Gustavo  CIV212  CON547
3 ...
...
```

El archivo está dividido en dos partes. La primera parte registra los cursos de la institución, en cada línea aparece el código del curso la cantidad de créditos y su nombre. No está registrado el número de cursos ni hay marcas que indiquen el fin de esta parte.

La segunda parte contiene un listado de matrícula, en cada línea aparece el nombre del 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. Las líneas están numeradas.

**PARTE 1: Lectura y almacenamiento de cursos (6 puntos)**

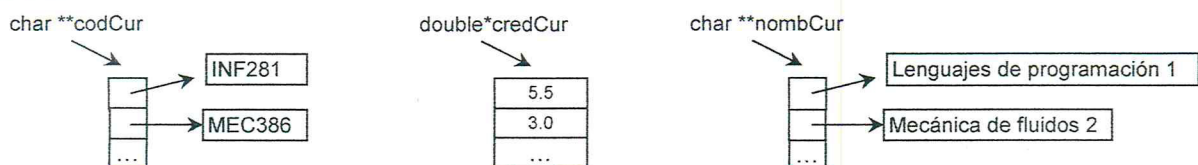
Debido a que la lectura y el almacenamiento de datos en un programa es una tarea muy importante, y que hay que entender que de nada sirve un programa hecho si los datos no se leyeron o almacenaron correctamente; **NO SE CALIFICARÁ LAS SIGUIENTES PARTES DEL LABORATORIO SI ES QUE ESTA PARTE NO ESTA TERMINADA O NO SE VERIFICA EL INGRESO CORRECTO DE LOS DATOS.**

**Proyecto:** Elabore un proyecto, denominado "*Parte1*", en la que el programa principal llame a las funciones siguientes:

Función **leeCursos** (codCur, credCur, nombCur);

Función **imprimeCursos** (codCur, credCur, nombCur);

La función **leeCursos** deberá leer, re direccionando la entrada estándar de datos, el archivo descrito anteriormente, sólo hasta la segunda parte, y guardar la información en las estructuras que se muestran a continuación:



Podrá emplear cualquier método para la asignación de datos pero al final de la lectura todas las estructuras tendrán asignados espacios de memoria exacta. No se pueden agregar argumentos a las funciones.

## PARTE 2: Lectura y confección de listas de clase (9 puntos)

De igual manera que en la parte 1; **NO SE CALIFICARÁ LAS SIGUIENTES PARTES DEL LABORATORIO SI ES QUE ESTA PARTE NO ESTA TERMINADA O NO SE VERIFICA EL INGRESO CORRECTO DE LOS DATOS.**

**Proyecto:** Elabore un proyecto, denominado "Parte2", en la que el programa principal, además de llamar a las dos funciones creadas en la parte 1, deberá llamar a las funciones siguientes:

Función `leerYCrearListasDeClases (listasDeClases, codCur)`

Función `imprimeListasDeClase (codCur, credCur, nombCur, listasDeClases);`

La función `leerYCrearListasDeClases` 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 nombres de los alumnos matriculados en ese curso.

Podrá emplear cualquier método para la asignación de datos pero al final de la lectura todas las estructuras tendrán asignados espacios de memoria exacta.

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

La función `imprimeListasDeClase` deberá generar el siguiente reporte:

```
Listas de clases:
Clave: INF281
Curso: Lenguajes de programación 1
Alumnos:
1.- Roncal/Neyra/Ana-Cecilia
2.- Lopez/Perez/Carlos
3.- Garcia/Castro/Pero-Jose-Carlos
...
Clave: MEC386
...
```

No se pueden agregar argumentos a las funciones.

## PARTE 3: Confección de carga de alumnos (5 puntos)

De igual manera que en la parte 1 y 2; **NO SE CALIFICARÁ LAS SIGUIENTES PARTES DEL LABORATORIO SI ES QUE ESTA PARTE NO ESTA TERMINADA O NO SE VERIFICA EL INGRESO CORRECTO DE LOS DATOS.**

**Proyecto:** Elabore un proyecto, denominado "Parte3", en la que el programa principal, además de llamar a las dos funciones creadas en la parte 1 y 2, deberá llamar a la funciones siguiente:

Función `imprimeCargaDeAlumnos (codCur, credCur, nombCur, listasDeClases);`

La función deberá leer una tercera parte en el archivo, para lo cual deberá agregar al final del archivo unos cuantos nombres de alumnos, luego de lo cual por cada alumno deberá imprimir el siguiente reporte empleando únicamente las estructuras creadas.

```
Alumno: Castro/Roca/Maria-Rosa
Codigo      Curso      Creditos
INF505      Graficos en computación  3.25
INF777      Inteligencia artificial  4.30
...
Total de créditos: 18.25
Alumno: ...
...
```

Al finalizar el laboratorio, comprima la carpeta *Lab3* 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 horario y aula.

**Profesores:** Miguel Guanira E.

**San Miguel, 4 de octubre del 2013.**