

TÉCNICAS DE PROGRAMACIÓN
SEXTO LABORATORIO
SEMESTRE ACADÉMICO 2020-2

Horario: Todos

Duración: 120 minutos

Elaborado por los profesores del curso.

ADVERTENCIAS:
<ul style="list-style-type: none"> - SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO. ESTO, Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA .c O .h CON FECHA U HORA DE CREACIÓN ANTERIOR AL LABORATORIO SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.

INDICACIONES:
<ul style="list-style-type: none"> - LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE, por lo que NO SE CALIFICARÁN aquellos módulos que son llamados por otros que estén incompletos. Cada módulo no debe sobrepasar las 30 líneas de código aproximadamente. - NO SE PUEDEN EMPLEAR ARCHIVOS AUXILIARES, VARIABLES GLOBALES NI <u>ARREGLOS</u>. - En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios. - DEBE COLOCAR SU NOMBRE Y CÓDIGO EN CADA UNO DE LOS ARCHIVOS .h Y .c QUE EMPLEE EN SUS PROYECTOS.

Cree un proyecto en NetBeans con el nombre: “ArchBin_Lab06_2020_2” y en él desarrolle el programa que resuelva el problema que se describe a continuación.

Una entidad educativa requiere de una aplicación en Lenguaje C para actualizar las notas de los alumnos al terminar un semestre académico. Para ello cuenta con la información en los siguientes archivos:

Alumnos.csv

```
5368,Perez/Ruiz/Soledad,2017,125.5,12
4521,Castro/Martinez/Alejandro/Martin,2015,150,45.5
7954,Rubio/Flores/Juan/Luis,2018,110
...
```

“Alumnos.csv” Contiene la lista de los alumnos de la entidad educativa. En cada línea contiene el código del alumno, su nombre y el año de ingreso, la cantidad de créditos aprobados y la cantidad de créditos desaprobados. Las últimas cantidades son las acumuladas hasta el semestre anterior. Si un alumno no tiene ningún curso desaprobado la línea respectiva no tiene el último dato (no considerar la posibilidad que un alumno no tenga cursos aprobados).

Notas.csv

```
102,5368,12.5,8767,11,7627,8, ...,1235,7.3
210,2210,3,6756,17.3,6778,12,7888,15.5, ...,6756,13.5
...
105,2189,5,6778,12.4,4523,10.4, ...,8912,18
...
```

“Notas.csv” Contiene las notas por cada curso que se obtuvieron en el último semestre académico. En cada línea se muestra el código del curso y la lista de los alumnos matriculados en el curso, cada alumno va acompañado de su respectiva nota. Esta lista, conformada por el código del alumno y la nota, es variable, debido a que en cada curso la cantidad de alumnos matriculados es diferente. Los datos no están ordenados.

Cursos.csv

```
5368,Calculo diferencial,4.5
2210,Laboratorio de quimica basica,2
..
2189,Matematicas discretas,5
...
```

“Cursos.csv” Contiene la lista de los cursos que brinda la entidad educativa. En cada línea se muestra el código del curso, el nombre y la cantidad de créditos. Los datos no están ordenados.

La institución educativa requiere un programa que emita un reporte para mostrar el consolidado de créditos y el detalle de las notas del último ciclo. El reporte debe ser similar al siguiente:

Resultados.txt

RESULTADOS ACADÉMICOS DE LOS ALUMNOS				
=====				
01) Código:4521 Nombre:Castro Martinez Alejandro Martin Ingreso:2017 Cred.aprobados:125.5 Cred.desaprobados: 12.0				

DETALLE DE NOTAS DEL ÚLTIMO SEMESTRE				
No.	CODIGO	NOMBRE	CREDITOS	NOTA

001	235	Geografia	3.50	11.2
002	102	Matematica Basica	5.00	13.5
003	121	Fisica 1	3.50	10.3
...
=====				

TOTAL DE ALUMNOS: 234				
PORCENTAJE DE ALUMNOS CON PROMEDIO PONDERADO APROBADO EN EL ÚLTIMO SEMESTRE: 78.37%				
PORCENTAJE DE ALUMNOS CON PROMEDIO PONDERADO DESAPROBADO EN EL ÚLTIMO SEMESTRE: 21.63%				

El proyecto deberá permitir realizar lo siguiente:

1. (3 puntos) Crear un archivo binario con el contenido del archivo “**Alumnos.csv**”. El archivo se denominará “**Alumnos.bin**” debe contener los mismos campos del archivo csv y adicionalmente los siguientes campos:

- Matricula (arreglos de estructuras “Cursos matriculados”), máximo 8 elementos.
- Cantidad de cursos matriculados (entero), el primer valor de este campo es cero.

La estructura “Cursos matriculados” debe contener los siguientes campos:

- Código del curso (entero)
- Nota (valor de punto flotante)

En un reporte muy simple, debe verificar la creación correcta del archivo, esta verificación debe hacerse en una función independiente y debe ser ejecutada fuera de la función que crea el archivo.

2. (3 puntos) Crear un archivo binario con el contenido del archivo “**Cursos.csv**”. El archivo se denominará “**Cursos.bin**” debe contener los mismos campos del archivo csv.

En un reporte muy simple, debe verificar la creación correcta del archivo, esta verificación debe hacerse en una función independiente y debe ser ejecutada fuera de la función que crea el archivo.

3. (6 puntos) Actualizar el archivo “**Alumnos.bin**” utilizando la información del archivo “**Notas.csv**”. Si existe una nota para un alumno que no existe en el archivo de alumnos, la nota se descarta. Debe actualizar los campos: matricula, cantidad de cursos matriculados, cantidad de créditos aprobados y la cantidad de créditos desaprobados.

4. (4 puntos) Actualizar el archivo “**Alumnos.bin**” ordenando en forma ascendente por el nombre del alumno.

5. (4 puntos) Imprimir el reporte “**Resultados.txt**”. El nombre del alumno debe mostrarse sin / . Si existe un alumno con nota en un curso, pero el curso no se encuentra en el archivo “**Cursos.bin**”, se debe mostrar el código del curso y la nota, sin descripción ni número de créditos.

Consideraciones:

- Las estructuras de datos que permita almacenar la información de los archivos deberán almacenar, los valores numéricos que aparecen en el archivo como tal, esto es, los enteros como int y los de punto flotante como double, los campos que manejan cadenas de caracteres deben definirse como tal.
- Los archivos CSV solo se pueden leer una vez.
- **NO podrá cargar los datos de los archivos en arreglos.**
- Toda operación de búsqueda debe realizarse en una función independiente. No se considerará en la calificación los procesos de búsqueda que estén contenidos en el código de otro proceso. Las funciones de búsqueda deben considerar la posibilidad que el dato buscado no se encuentre.

ADVERTENCIAS:

- Obligatoriamente debe desarrollar su proyecto bajo NetBeans en Windows, no podrá desarrollarlo empleando otro IDE ni otro sistema operativo.
- Al finalizar el laboratorio comprima la carpeta de su proyecto empleando **el programa Zip que viene por defecto en el Windows**, no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares. Luego súbalo a la tarea programa en Paideia para este laboratorio.

CRITERIOS DE CALIFICACIÓN:

1. Si el programa presentado tiene más de tres errores de sintaxis serán calificados sobre la mitad del puntaje.
2. Si el programa no muestra los resultados o los que muestra no sean correctos, no podrán tener más del 75% de la nota.
3. Se descontará 15% de la nota si el programa define variables con nombres que no tengan sentido. Las variables deben empezar con una minúscula, se emplearán mayúsculas para separar las palabras compuestas (p. e.: baseInf).
4. Se descontará 15% de la nota si no se colocan comentarios relevantes, incluyendo un encabezado al inicio del programa en el que se indique el nombre del autor, la fecha, y una descripción de lo que hace el programa).
5. No se calificará el código puesto como comentario.
6. No se calificarán aquellas funciones implementadas en el archivo main.c

San Miguel, 15 de diciembre del 2020