

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**LENGUAJES DE PROGRAMACIÓN 1**

**5ta práctica (tipo b)**  
**Primer Semestre 2021**

**Indicaciones Generales:**

- Duración: 110 minutos.

Obligatoriamente los alumnos deberán mantener en todo momento el AUDIO Y VIDEO de sus computadores abierto de modo que puedan recibir los comunicados que se hagan durante el examen y la revisión de los trabajos que estén desarrollando. De tener algún problema deberán hacérselo saber de inmediato al profesor de su horario por correo. De no hacerlo, no se aceptarán reclamos alegando que no oyeron las indicaciones.

- No se pueden emplear variables globales, estructuras, ni objetos (con excepción de los elementos de iostream, iomanip y fstream). Tampoco se podrán emplear las funciones malloc, realloc, strdup o strtok, igualmente no se puede emplear cualquier función contenida en las bibliotecas stdio.h, cstdio o similares y que puedan estar también definidas en otras bibliotecas. Tampoco podrá hacer uso de plantillas.
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. Cada función NO debe sobrepasar las 20 líneas de código aproximadamente. El archivo **main.cpp** solo podrá contener la función main de cada proyecto y el código contenido en él solo podrá estar conformado por tareas implementadas como funciones. 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.
- NO SE CALIFICARÁN aquellas funciones desarrolladas en el mismo archivo que la función main.
- El código comentado NO SE CALIFICARÁ.
- Los proyectos deben obligatoriamente desarrollarse en NetBeans bajo el sistema operativo Windows. No se revisarán los proyectos desarrollados en otros sistemas operativos o IDEs.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no muestren resultados o que estos no sean coherentes en base al 60%.  
Se tomará en cuenta en la calificación el uso de comentarios relevantes.

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 .cpp O .h CON FECHA U HORA DE CREACIÓN ANTERIOR A LA EVALUACIÓN SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.

**NO SE HARÁN EXCEPCIONES ANTE CUALQUIER TRASGRESIÓN DE LAS INDICACIONES  
DADAS EN LA PRUEBA**

- **Puntaje total:** 20 puntos.

**Cuestionario:**

La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en el capítulo 5 del curso: "Arreglos y punteros". En este laboratorio se trabajará con memoria dinámica y el uso de punteros múltiples. Se emplearán los métodos de asignación exacta y por incrementos.

Deberá elaborar un proyecto denominado "LAB05\_PuntMultiples" y en él desarrollará el programa que dé solución al problema planteado. DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCOTARÁ 3 PUNTOS DE LA NOTA FINAL.

Se tienen tres archivos del tipo CSV, los cuales se describen a continuación:

Conductores.csv
30018447,FUENTES QUISPE CINTHIA DELIA
32517791,ZORRILLA LARA ARTURO
...

licencia,nombre del conductor

RegistroDeFaltas.csv
30018447,O1C-880,25/8/2020,102
78153392,A7R-205,18/12/2017,672
...

licencia,placa,fecha,infracción

Infracciones.csv
103,Detener el vehículo bruscamente sin motivo,Grave,316.00
207,Utilizar la bocina para llamar la atención en forma innecesaria,Leve,158.002
...

infracción,descripción,gravedad, monto

Con esta información, la función "main" del proyecto estará compuesto por el siguiente código:

```
#include "BibManejoDinamicoDeMemoria.h"
int main(int argc, char** argv) {
    char **conductor, ***placa;
    int *licencia, **falta;
    double **multa;
    cargarConductores(licencia, conductor);
    reporteConductores(licencia, conductor);
    cargarFaltas(licencia, placa, falta);
    reporteFaltas(licencia, conductor, placa, falta);
    resumenMultas(licencia, falta, multa);
    reporteMultas(licencia, conductor, multa);
    return 0;
}
```

**NO PUEDE  
CAMBIAR  
ESTE CÓDIGO**

### Pregunta 1 (4 puntos)

Implemente las funciones **cargarConductores** y **reporteConductores**, la primera debe leer las licencias y nombres y nombres de los conductores, desde el archivo CSV, y colocar los datos en las estructuras representadas en la figura No. 1 según corresponda. Los espacios de memoria asignados deberán ser **dinámicos y por incrementos de 5 en 5 en todos los casos** (salvo las cadenas que deben ser exactas). La función **reporteConductores** tiene la finalidad de verificar que los datos hayan sido correctamente colocados en la estructura, los datos deben aparecer alineados correctamente con encabezados apropiados que indiquen la naturaleza de los datos (no podrá emplear el carácter '\t').

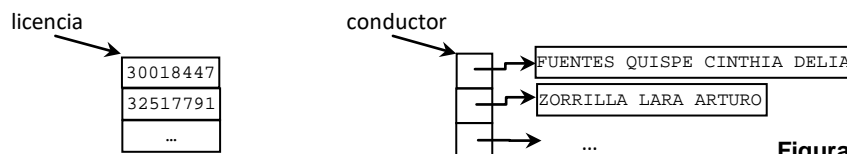


Figura No. 1

### Pregunta 2 (10 puntos)

**Esta pregunta no tendrá sentido si no implementa la Pregunta 1**

Implemente las función **cargarFaltas** y **reporteFaltas**, la primera debe cargar los datos de las faltas cometidas del archivo CSV, y colocarlos en las estructuras como se muestra en la figura No. 2, siguiendo el orden de licencias de la pregunta anterior. **El archivo CSV debe se leerse una sola vez**, en todo el programa. Los espacios de memoria asignados para los arreglos primarios y a las cadenas de caracteres deben ser **dinámicos y exactos**, el resto de los arreglos deben ser asignados de manera **dinámica y por incrementos de 5 en 5**. La función **reporteFaltas** tiene la finalidad de verificar que los datos hayan sido correctamente colocados en la estructura, los datos deben aparecer alineados correctamente con encabezados apropiados que indiquen la naturaleza de los datos (no podrá emplear el carácter '\t').

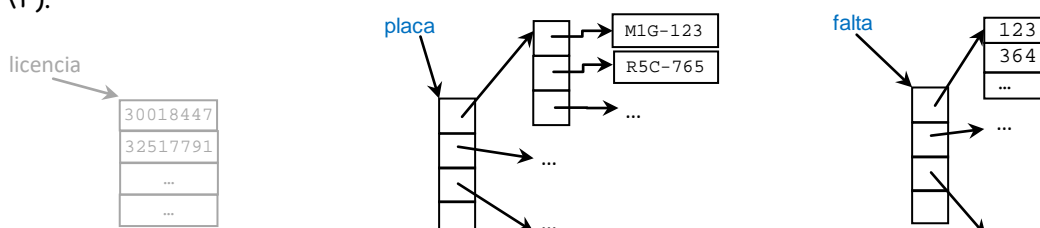
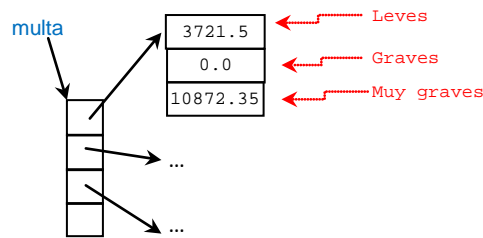


Figura No. 2

**Pregunta 3 (6 puntos)**

**Esta pregunta no tendrá sentido si no implementa la Pregunta 1 y 2**

Implemente las funciones **resumenFaltas** y **reporteMultas**, la primera deberá trabajar con el archivo de infracciones para determinar el total que debe pagar cada conductor por sus faltas leves, graves y muy graves, estos valores serán colocados en una estructura como la mostrada en la figura No. 3. La función **reporteMultas** tiene la finalidad de verificar que los datos hayan sido correctamente colocados en la estructura, los datos deben aparecer alineados correctamente con encabezados apropiados que indiquen la naturaleza de los datos (no podrá emplear el carácter '\t').



Al finalizar la práctica, 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.

Profesores del curso: Miguel Guanira  
Rony Cueva

San Miguel, 14 de mayo del 2021.