

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

**LENGUAJES DE PROGRAMACIÓN 1**

**4ta 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 método de asignación exacta de memoria.

Deberá elaborar un proyecto denominado "Laboratorio04-07-05-2021" y en él desarrollará el programa que dé solución al problema planteado. DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 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 "BibliotecaMemoriaDinamica.h"
int main(int argc, char** argv) {
    char **conductor,*gravedad;
    int *licencia,*infraccion;
    double *multa;

    cargarConductores(licencia, conductor);
    reporteConductores(licencia, conductor);
    cargarInfracciones(infraccion, gravedad, multa);
    reporteInfracciones(infraccion, gravedad, multa);
    reporteDeFaltas(licencia, conductor, infraccion, gravedad, multa);
    return 0;
}
```

**NO PUEDE  
CAMBIAR  
ESTE CÓDIGO**

### Pregunta 1 (4.0 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 exactos en todos los casos**. 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').

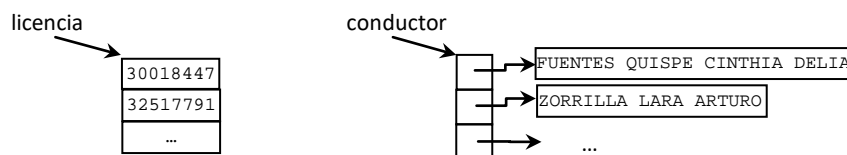


Figura No. 1

### Pregunta 2 (4.0 puntos)

Implemente las funciones **cargarInfracciones** y **reporteInfracciones**, la primera debe leer los datos del archivo de infracciones y colocarlos en las estructuras representadas en la figura No. 2 según corresponda. Los espacios de memoria asignados deberán ser **dinámicos y exactos en todos los casos**. La función **reporteInfracciones** 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').

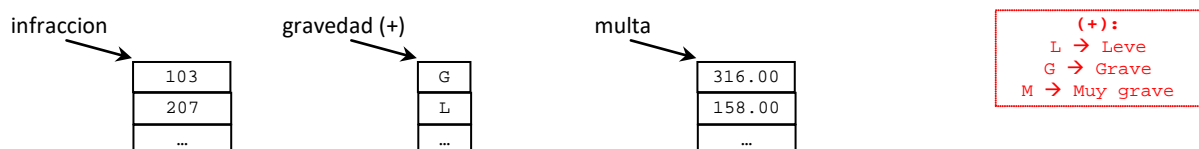


Figura No. 2

### Pregunta 3 (12 puntos) Esta pregunta no tendrá sentido si no implementa las preguntas 1 y 2.

Implemente la función **reporteDeFaltas**. Esta función trabajará con el archivo que contiene el registro de faltas, así como con las estructuras que recibe como parámetros. La función deberá emitir un reporte como el que se indica a continuación:

FALTAS LEVES				
No.	Licencia	Placa	Cantidad	Monto
1)	30018447	O1C-880	2	245.75
2)	78153392	A7R-205	4	902.56
...	...	...	...	...
53)	30018447	H2R-234	1	125.35
*****				
FALTAS GRAVES				
No.	Licencia	Placa	Cantidad	Monto
1)	...	...	...	...
...	...	...	...	...
*****				
FALTAS MUY GRAVES				
No.	Licencia	Placa	Cantidad	Monto
1)	...	...	...	...
...	...	...	...	...
*****				

El proceso para desarrollar el reporte se seguirá estrictamente los siguientes pasos:

- Leer uno a uno los datos del archivo **reporteDeFaltas** y con solo las faltas "LEVES" deberá construir las estructuras representadas en la figura No. 3. **En este paso solo podrá recorrer el archivo una sola vez.** La estructura no podrá contener placas repetidas, por lo que deberá acumular las cantidades y montos. **Los espacios de memoria deberán ser dinámicos y exactos en todos los casos.**

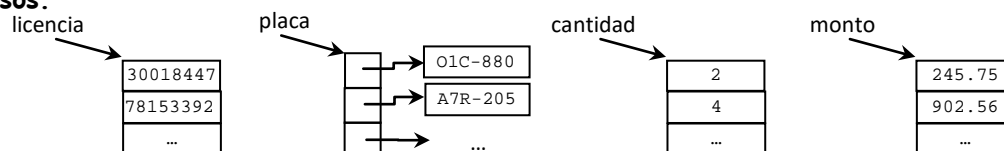


Figura No. 3

- Con las estructuras armadas deberá elaborar el reporte solo para la parte de **faltas leves**.
- Liberar todos los espacios de memoria asignados a estas cuatro últimas estructuras.
- Repetir los tres primeros pasos anteriores para las faltas graves.
- Repetir los tres primeros pasos anteriores para las faltas muy graves.
- Al final de los tres procesos el archivo se debió recorrer tres veces, una por cada tipo de falta.

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, 7 de mayo del 2021.