Tarea 2

Manejo de datos y punteros

Entrega: viernes 5 de mayo a las 21:00 hrs.

1. Introducción

Al momento de sacar el permiso de circulación para su auto muchas personas se dan cuenta de que tienen multas asociadas que les impiden completar el trámite. Esto genera molestia en los conductores que exigen poder tener una herramienta online donde ver sus deudas antes de iniciar el pago del permiso.

La municipalidad de Pelotillehue le ha contratado para desarrollar esta plataforma, para ello le proporcionan la información de todos los vehículos inscritos en las distintas comunas y otro archivo que contiene todas las patentes de vehículos y los montos de sus duedas. Le piden que pueda calcular la deuda total de cada vehículo, además de sacar estadísticas específicas según criterios como modelo, tipo de vehículo, comuna en la cual están inscritos, etc.

2. La Tarea

Para resolver esta tarea, se debe implementar un programa en lenguaje C que permita cargar la información de ambas bases de datos y filtrar la información basándose en los parametros solicitados. Se proporcionará como entrada 2 archivos que contenga la información de la base de datos de vehículos inscritos y el archivo de multas.

En primer lugar, el programa debe leer los archivo de entrada "vehiculos.txt" y "multas.txt" almacenar la información de los vehículos y de sus deudas, esta información estará en el formato `patente,modelo,tipo,marca,color,coumna` y `patente,monto,pagada`, respectivamente.

Además es necesario guardar los vehículos en un arreglo en forma de `struct Vehiculo` que contenga los parámetros del vehículo para luego poder ser referenciados a la hora de trabajar con la información.

Su programa será controlado por el usuario a través del *standard input* mediante una sucesión de comandos. Los comandos a implementar y sus respectivas funcionalidades son los siguientes:

- `deuda [patente]`: el programa deberá indicar por la consola la deuda total por concepto de multas impagas que tenga el vehículo de la patente indicada. Las patentes tendrán 5 a 6 caracteres, consistentes en dos a cuatro letras seguidas por la cantidad restante en dígitos numéricos.
- `deuda [n]`: el programa deberá listar las n patentes con más deudas impagas, con sus respectivos montos.

- `deudores comuna [comuna]`: el programa deberá entregar en la consola las patentes con deudas en la comuna, y el monto adeudado por cada una, ordenados desde el mayor al menor deudor.
- `deudores patente [string]`: el programa deberá listar los deudores cuyas patentes comiencen con el string indicado, y el monto adeudado por cada uno, ordenados alfabéticamente. Por ejemplo, si el usuario pide `deudores patente EL5`, deberá buscar todas las patentes desde `EL500` a `EL5999`.
- `salir`: para terminar la ejecución del programa. Mientras no se utilice este comando, el programa deberá volver a esperar comandos por parte del usuario luego de terminar cada tarea.

Su programa será evaluado con un acento especial en el tiempo de ejecución de múltiples consultas, por lo que es recomendable el paso de parámetros por referencia y mantener los datos preordenados de más de una forma, según sea necesario. Para ello, debe leer los dos archivos en su totalidad y mantenerlos en memoria mientras el usuario ejecute consultas. De manera secundaria, se valorará el uso eficiente de la memoria, por lo que se recomienda que mantenga una sola copia de cada `struct` y sólo repita las referencias (punteros).

3. Consideraciones Generales

Su programa debe ser robusto frente a datos que no corresponden. Esto quiere decir que su programa no debe caerse en caso de que el usuario haga operaciones matemáticas no válidas (p.ej., dividir por cero si fuera el caso). Sin embargo, se puede asumir que el usuario será amigable, en el sentido de que cuando le soliciten un número este no ingresará letras, por ejemplo.

Recuerde que el archivo puede tener más de una línea de texto.

4. Evaluación y Entrega

Esta tarea puede ser resuleta en grupos de máximo 2 personas. El plazo para la entrega de la tarea vence impostergablemente el viernes 5 de mayo a las 21:00 hrs..

Formato de entrega: Subir un único archivo con el código de su programa al módulo de tareas de la página del curso, con un nombre de archivo que incluya ambos apellidos de los autores de la tarea separados por guiones, seguido de "Tarea2.c", de la forma "`Prat Chacon-Carrera Pinto-Tarea2.c`". El incumplimiento de este requisito de nombre de archivo será penalizado con un descuento de un punto en la nota. Los archivos compilados no serán tomados en cuenta, si se llega a subir solo un archivo compilado, este será ignorado, y será evaluado con nota 1.

En el momento de compilación, se deberá indicar las siguientes *flags* al compilador:

```
-Wall -Wextra -Wundef -Werror -Wuninitialized -Winit-self
```

Aquellas tareas que no compilen de la forma básica (`gcc -o salida entrada`) serán evaluadas con nota 1. Las que compilen, pero se caigan durante la ejecución, serán evaluadas con nota máxima 3.

Su programa será evaluado con múltiples casos de prueba y deberá ser capaz de ejecutarlos todos de manera correcta. De fallar en algún caso de prueba serán descontados los puntos correspondientes a dicho caso.

5. Consideraciones de Trabajo

El trabajo en esta tarea puede hacerse en parejas. Cuide su tarea para que no sea copiada parcial o íntegramente por otros. Todas las tareas entregadas serán comparadas por un sistema automático de detección de plagios. Cualquier copia (de otras tareas o de internet) será penalizada, recibiendo el mismo castigo tanto quien copia como quien permite que le copien. También es considerada copia cualquier ayuda externa recibida directamente en la tarea, sin importar si proviene de un alumno del curso, de la universidad, o de otro lugar. El castigo será establecido por el Consejo de la Facultad, siendo como mínimo un 1,0 de promedio en el curso.