



# Solemne 1

Profesores: Juan Ricardo Giadach, Roberto Konow, Leandro LLanza, Cristian Tala

Tiempo: 110 minutos

**Recuerde que sólo puede utilizar apuntes personales e intransferibles.**

## Pregunta 1 (30 puntos)

Para encontrar la cercanía literal de dos documentos se puede utilizar la técnica que consiste en buscar las palabras que comparten esos documentos, o encontrar la frecuencia de ciertas palabras dentro de ellos.

Vamos a considerar a un documento como una lista enlazada simple de palabras, y a una biblioteca, como un conjunto de documentos.

Asumiendo que la estructura de datos Documento (que utiliza una lista enlazada simple para representar un documento cualquiera) tiene los siguientes métodos ya implementados.

`Insertar(Palabra s, int i) // Inserta la palabra s en la posición i.`

`Borrar(int i) // Borrar la palabra de la posición i.`

Se le pide

a) Definir las estructuras Biblioteca, Documento y Palabra (sólo definir)

e implementar los siguientes métodos de la clase Biblioteca:

b) `void intersectar(Documento x, Documento y)`

que recibe como parámetros dos documentos e imprime en pantalla todas las palabras que comparten estos dos documentos.

c) `int frecuencia(Palabra s)`

que retorna la cantidad de veces que aparece s en todos los documentos de la biblioteca.

Nota: Puede agregar e implementar los métodos o funciones que estime conveniente.

## Pregunta 2 (30 puntos)

Al aproximarse el fin de cada mes, las filas de vehículos esperando ingresar a las plantas de revisión técnica se incrementan. Como ya es sabido, es el último dígito de la patente el que determina el mes en que se debe hacer tal revisión. Sin embargo, muchos automovilistas no cumplen con esta obligación en el mes que les corresponde y van a la planta de revisión técnica uno o más meses después, lo que alarga las colas y los tiempos de espera.

La empresa *PRT Express* ha decidido favorecer a quienes están haciendo la revisión en el mes que les corresponde y ha dispuesto que 4 de sus 6 puestos de revisión serán destinados a ellos, mientras que los rezagados serán atendidos en los 2 puestos restantes. Obviamente, si alguno de los puestos de revisión queda desocupado (por no haber vehículos de la categoría esperando) puede atender a un vehículo de la otra categoría.

Para mejorar su servicio, la empresa requiere una aplicación que permita informar el tiempo de espera que tendrá una persona cuando llegue a hacer la fila, asumiendo que el tiempo de revisión es el mismo para cada vehículo.

Se pide hacer una función que reciba la patente de un vehículo y retorne los minutos que deberá esperar antes de comenzar a ser atendido en alguno de los puestos de la planta. No olvide definir todas las estructuras de datos que utilice.

## Pregunta 3 (bonus track 10 puntos)

Mi vieja impresora necesita imprimir un documento por ambas carillas, pero no cuenta con esa funcionalidad de fábrica.

Sabiendo que los documentos son enviados a la impresora en una lista enlazada (de enlace simple) he googleado el problema y he obtenido la siguiente solución.

*Imprimir primero sólo las páginas impares, colocar esas páginas de nuevo en la impresora (pero al revés), esperar una confirmación del usuario e imprimir las pares.*

Por ejemplo, si un documento tiene 10 páginas, primero imprimo la 1, 3, 5, 7 y 9; tomo estas 5 hojas y las devuelvo a la impresora, presiono *enter* y se imprimen la 2, 4, 6, 8 y la 10.

Usando la estructura `Page`, implemente el método

```
void imprime2caras(Page x)
```

que resuelve este algoritmo en forma recursiva.

```
class Page {
    Data data;
    Page *next;
}
```

Para imprimir una página al papel, use el método `void print(Data x)`