

# Algoritmos y Estructuras de Datos

Examen Final 25/9/2024

## Ejercicio de Práctica en Papel

Nombre y Apellido: .....

Año de Cursado:.....

Nombre y Apellido: ..... Comisión:.....

*El objetivo de este ejercicio es evaluar la aplicación integral de contenidos. Para esto, evaluaremos la solución algorítmica desarrollada en base a la información definida en la consigna, la forma en la que diseñas y escribes código C++, y la aplicación de buenas prácticas de programación. El puntaje final obtenido corresponderá al global de la solución entregada. No se califica por incisos.*

*Todas las entregas deben tener nombre y apellido. Tener en cuenta la legibilidad de la solución entregada, ya que de no comprenderse lo escrito no se podrá corregir. Deben entregarse todas las hojas que contengan código asociado a la solución.*

Tiempo de Resolución: 90 minutos.

Puntaje Requerido: 20/40 puntos.

### Consigna:

La empresa deportiva BRUMA desarrolló un software para el registro y seguimiento de los envíos de productos comprados en su tienda online. La información de los **envíos pendientes de entrega** se encuentra almacenada en una lista dinámica definida en base al siguiente registro:

```
struct Envio {  
  
    int numero; //numero de seguimiento.  
  
    Paquete paq; //información del paquete a ser entregado.  
  
    Informacion origen, destino; //información del remitente y destinatario.  
  
};
```

Se sabe que para cada elemento de tipo *Paquete*, se considera relevante su tamaño (ancho, largo y alto, todos en cm), si el contenido es frágil o no (valor booleano) y el peso (en Kg). Por otro lado, para los campos de tipo *Informacion* se almacena nombre y apellido (tipo String), DNI (tipo entero) y *Dirección* (calle, número, ciudad y provincia).

Tomando como base estas definiciones, se solicita:

**a)** Definí todas las estructuras de datos que consideres necesarias para implementar la lista de **envíos pendientes de entrega**. Haciendo uso de tus definiciones, crea una lista vacía local a la función `main()`.

**b)** Cada cartero a cargo de los *Envíos*, reparte *Paquetes* de un mismo tipo. Los *Paquetes* en lista son clasificados según su tamaño en Grandes, Medianos, o Pequeños. La siguiente tabla muestra la clasificación de los paquetes según su volumen:

Clase	Volumen (cm <sup>3</sup> )
G	>= 125000
M	entre 15625 y 125000
P	<= 15625

## Algoritmos y Estructuras de Datos

Examen Final 25/9/2024

Definí e implementá una función que, tomando como argumento la lista de envíos pendientes y una clase de paquetes (definida como char), genere y devuelva la sublista de *Envíos* que contienen *Paquetes* de la clase indicada. Es decir, la lista que devuelve la función (Lista1) indica los *Envíos* que el cartero debe repartir según la clase indicada ('G','M' ó 'P'). Los *Envíos* incorporados en la sublista, no deben removerse de la lista original.

c) A medida que el cartero realiza el reparto, el contenido de Lista1 es actualizado. **Codifique** una función *RepartirCartas()* que reciba Lista1 y cada vez que el cartero realiza una entrega, invoque a la función *eliminarEnvio()* (**no es necesario codificarla**) para dicha actualización. La función *eliminarEnvio()* recibe la lista de envíos pendientes de entregar por el cartero y el número de seguimiento del *Envío* entregado. Como resultado de ejecutar esta función, la lista de envíos pendientes se actualiza removiendo el *Envío* entregado. Indique la complejidad de la función *RepartirCartas()* y de la función *eliminarEnvio()*.

d) Cuando el cartero regresa a la central (luego del reparto), aún pueden quedar *Envíos* en *LISTA1*, ya que no siempre los destinatarios están en casa. Definí e implementá una función *enviosNoEntregados()* que reciba como parámetro la lista resultante luego del reparto y retorne la cantidad de envíos no entregados. Esta función deberá ser implementada de forma recursiva.

e) Se dispone de dos vectores, *Remitentes* y *Destinatarios*, con sus respectivos tamaños lógicos, cuyo tipo base es *Informacion*, que tienen cada uno la lista completa de remitentes y destinatarios que participaron de algún envío en los últimos 10 años de la empresa. Ambos vectores están ordenados ascendentemente por DNI. **Codifique** una función que reciba los dos vectores y sus tamaños lógicos y muestre por pantalla un listado -ordenado por DNI- con apellido y nombre de todos los clientes Premium de la empresa (aquellos que son tanto Remitentes como Destinatarios). Indique que complejidad tiene su algoritmo solución.

**Importante:** Para la resolución del problema puedes codificar todas las funciones que consideres necesarias. Los campos de las estructuras de datos deben respetar lo enunciado en la consigna. En los casos donde no se indica un prototipo explícito en la consigna, los parámetros formales de funciones (cantidad y tipo) deben definirse según los objetivos propuestos. El puntaje final obtenido tendrá en cuenta la eficiencia de la estrategia de resolución elegida.