

Prueba de cátedra 2

INF2223 – Estructura de datos
I Semestre 2023

Información de la evaluación		Nota
Nombre y apellido:		
Asignatura:	INF 2223 - Estructura de Datos	
Fecha:	4 de mayo de 2023	

1. Enunciado

La banda criminal "Los Care'Nodo" ha decidido implementar un sistema de información para gestionar el control de los robos que realiza a bancos del país, ya que ha aumentado el número de participación delictual y requiere un control estricto de lo hurtado. Para ello se cuenta con:

- La Banda cuenta con un nombre, que en este caso es "Los Care'Nodo", un puntero a una lista simplemente enlazada circular de referencia a los ladrones que componen la banda y una referencia a los bancos que existen en el país, los cuales son gestionados a través de una lista doblemente enlazada con nodo fantasma.
- Cada Banco cuenta con nombre, dirección y saldo en caja fuerte.
- Un Ladrón tiene apodo, rut, y un arreglo unidimensional compacto de tamaño maxRobos con pLibre que referencia a los robos realizados por el ladrón.
- Un Robo tiene fecha, monto robado y una referencia al banco que se robó.

```
struct Banco{
    char *nombre;
    char *direccion;
    int saldo;
};

struct NodoBanco{
    struct Banco *datosBanco;
    struct NodoBanco *sig, *ant;
};

struct Robo{
    char *fecha;
    int monto;
    struct Banco *bancoRobado;
};

struct Ladron{
    char *apodo;
    char *rut;
    int maxRobos;
    struct Robo **robos;
    int pLibreRobos;
};

struct NodoLadron{
    struct Ladron *datosLadron;
    struct NodoLadron *sig;
};

struct BandaCriminal{
    char *nombre;
    struct NodoLadron *headLadrones;
    struct NodoBanco *headBancos;
};
```

Se pide implementar las siguientes funciones:

a. `struct Ladron **quitarLadronesMenosRentables(struct BandaCriminal *LosCareNodo)`

Recibe una referencia al sistema BandaCriminal y debe quitar del sistema a aquellos criminales que tienen un monto de robos acumulados bajo el promedio de todos los montos de robos acumulados de todos los clientes. Debe retornar un arreglo compacto de tamaño exacto con los ladrones quitados. En caso de fracaso retorna NULL.

Importante:

- Debe modularizar las soluciones e implementar lo solicitado.
- Comente los supuestos
- Respete las reglas de programación y desarrollo.

Prueba de cátedra 2

INF2240 – Estructura de datos
I Semestre 2022

Información de la evaluación		Nota
Nombre y apellido:		
Asignatura:	INF 2240 - Estructura de Datos	
Fecha:	6 de mayo de 2022	

1. Enunciado

En razón a la incertidumbre en la economía global producto de los efectos pandemia y en la economía nacional producto de las iniciativas de la convención constitucional, se ha producido un aumento acelerado de personas naturales que han procedido a colocar sus dineros en fondos de inversión. Para ello se tendrá un sistema que gestionará los fondos de inversión, los inversionistas y las compañías que administran dichos fondos, para esto se define:

- **Sistema Inversiones:** contiene una referencia a una lista simplemente enlazada circular que administrará las Administradoras de Inversiones y una referencia a una lista doblemente enlazada con nodo fantasma que gestionará a los Inversionistas.
- **Administradora:** se encarga de administrar e invertir los fondos de sus clientes, contiene un rol único tributario, nombre, nombre del gerente y una clasificación de riesgo.
- **Inversionista:** tiene nombre, rut y una colección unidimensional de tamaño `maxInv` de referencias a los Fondos invertidos, el cual corresponde a las inversiones de los clientes. Considere que un inversionista puede tener distintos fondos en distintas administradoras.
- **Fondo:** tiene un id, una fecha en la cual se realizó la inversión, el monto invertido y una referencia de la administradora que administra el monto invertido.

```
#define maxInv 4000

struct SistemaInversiones {
    struct NodoAdministradora *headAdmin;
    struct NodoInversionista *headInv;
};

struct NodoAdministradora {
    struct Administradora *detalle;
    struct NodoAdministradora *sig;
};

struct Administradora {
    int rolUnicoTributario;
    char *nombre;
    char *nombreGerente;
    float clasificadorRiesgo;
};

struct NodoInversionista {
    struct Inversionista *datosInversionista;
    struct NodoInversionista *ant, *sig;
};

struct Inversionista {
    char *rut;
    char *nombre;
    struct Fondo **inversiones;
};

struct Fondo {
    int id;
    char *fecha;
    int monto;
    struct Administradora *administradora;
};
```

Se pide implementar las siguientes funciones:

- a. `struct Administradora *quitarAdministradoraRiesgosa(struct SistemaInversiones *SI, float tasaInteres)`

Esta función recibe como argumento una referencia al sistema de inversiones y el valor de la tasa de interés dada por el Banco Central, y debe quitar la administradora de fondos con el mayor índice global de riesgo. Para determinar la administradora con el mayor índice global de riesgo deberá utilizar la formula indicada posteriormente. Debe retornar una referencia a la Administradora quitada y en caso de empate deberá quitar solo

la primera. Antes de eliminar la Administradora deberá mover los fondos de los inversionistas que tenían dineros asociados a dicha Administradora, para ello, deberá reasignar todos los fondos de esa administradora a la que tenga el menor índice global de riesgo. En caso de fracaso o que no se pueda ejecutar una acción no quita la Administradora y retorna NULL. La fórmula para calcular el índice global de riesgo es:

$$IGR = \sum_{i=1}^n \left(\left(\sum_{f=1}^{maxInv} Monto_{fi} \right) * ClasificadorRiesgo_i \right) * \frac{1}{tasaInteres}$$

Donde:

- $Monto_{fi}$: corresponde al monto invertido por el inversionista en dicha administradora.
- $ClasificadorRiesgo_i$: corresponde al clasificador de riesgo de la administradora.

Importante:

- Procure modularizar las soluciones e implementar lo solicitado.
- Comente los supuestos
- Respete las reglas de programación y desarrollo.
- No puede alterar las funciones entregadas ni sus indicaciones de diseño, hacerlo implica nota mínima.

Prueba de cátedra 2

INF2223 – Estructura de datos
II Semestre 2023



Información evaluación		Nota
Nombre y apellido:		
Asignatura:	INF2223 – Estructura de datos	
Fecha:	Jueves, 28 de septiembre de 2023	

Enunciado

El sistema de Transporte Metropolitano de Valparaíso (TMV) organiza su estructura de registro de viajes de la siguiente forma:

- El **Sistema de Transporte** contiene una referencia hacia una lista simplemente enlazada que organiza los buses que tiene el sistema y una referencia a una lista doblemente enlazada circular que contiene a los pasajeros existentes.
- Los **Buses** tienen una patente, marca, modelo, y capacidad total.
- Los **Pasajeros** tiene número id, nombre, y un arreglo compacto con el uso de pLibre de referencias a los buses con los viajes programados que tiene.

```
#include <stdio.h>
#include <stdlib.h>
#define cantViajes 1000

struct Bus{
    char *patente;
    char *marca;
    char *modelo;
    int capacidadTotal;
};

struct NodoBus{
    struct Bus *datosBus;
    struct NodoBus *sig;
};

struct Pasajero{
    int id;
    char *nombre;
    struct Bus **viajes;
    int pLibre;
};

struct NodoPasajero{
    struct Pasajero *datosPasajero;
    struct NodoPasajero *ant, *sig;
};

struct SistemaTransporte{
    struct NodoBus *headBuses;
    struct NodoPasajero *headPasajeros;
};
```

Se solicita:

```
struct Bus *quitarBusAveriado(struct SistemaTransporte *TMV, char *patenteBus)
```

Esta función recibe una referencia al sistema de Transporte y la patente de un bus que debe ser quitado por desperfecto técnico. El procedimiento para reparar un bus se debe reparar a través de la compañía de seguros, lo cual demora aproximadamente 10 días; en caso contrario el mismo sistema puede repararlo en cuestión de hora, pero debe asumir el costo de dicha reparación, por lo cual no es la opción seleccionada siempre. Para quitar el bus debe reasignar todos los pasajeros que tienen viajes programados en dicho bus a otros buses que tengan el mismo destino y tengan capacidad disponible. En caso de no encontrar buses para todos los pasajeros del bus a eliminar entonces los pasajeros no se deberán reasignar y el bus no se quita. Para determinar si un bus tiene capacidad disponible deberá comparar la variable capacidadTotal del bus con la cantidad de referencias existentes desde los pasajeros a dicho bus.

Consideraciones obligatorias

- Debe modularizar las soluciones e implementar lo solicitado.
- Comente los supuestos en caso de ser necesario
- Respete las reglas y buenas prácticas de programación y desarrollo.

Prueba de cátedra 2

INF2223 – Estructura de datos
I Semestre 2024

ESCUELA DE
INGENIERÍA INFORMÁTICA



PONTIFICIA
UNIVERSIDAD
CATÓLICA DE
VALPARAÍSO

Información evaluación		Nota
Nombre y apellido:		
Asignatura:	INF2223 – Estructura de datos	
Fecha:	Miércoles, 8 de mayo de 2024	

Enunciado

La Fórmula 1 necesita un sistema que le ayude a gestionar más rápidamente sus componentes organizacionales. Para ello se ha implementado un sistema que cuenta con las siguientes estructuras:

- *Formula 1* tiene una referencia a una lista simplemente enlazada con nodo fantasma que organiza las Scuderías. Una scudería es un equipo o escudería de competición y están formadas por personal técnico, mecánicos, ingenieros, pilotos, etc. cuya misión es preparar uno o varios vehículos de competición y participar en campeonatos o carreras. Además, se cuenta con una referencia a una lista circular doblemente enlazada de pilotos existentes en la Fórmula 1.
- Cada *Piloto* tiene un id (único en todo el sistema), nombre, edad y sueldo (en dólares).
- La *Scudería* tiene nombre, fecha de creación, la cantidad de campeonatos ganados y una referencia a un struct que contiene los datos del equipo que la compone.
- El *Equipo* tiene el nombre del técnico jefe, la cantidad de técnicos, un arreglo de referencias a los técnicos del equipo (el tamaño del arreglo es el indicado como cantidad de técnicos, no necesariamente está lleno) que además cuenta con el manejo de pLibre y un arreglo de tamaño 3 que referencia a los pilotos que representan a la Scuderia en los campeonatos (2 pilotos oficiales y 1 de reemplazo).
- El *Técnico* tiene nombre, edad y especialidad.

```
struct Piloto{
    int id;
    char *nombre;
    int edad;
    float sueldo;
};

struct NodoPiloto{
    struct Piloto *datosPiloto;
    struct NodoPiloto *ant, *sig;
};

struct Tecnico
{
    char *nombre;
    int edad;
    char *especialidad;
};

struct Equipo
{
    char *nombreJefe;

    int cantidadTecnicos;
    struct Tecnico **tecnicos;
    int pLibreTecnicos;
    struct Piloto *pilotos[3];
};

struct Scuderia{
    char *nombre;
    char *fechaCreacion;
    int campeonatosGanados;
    struct Equipo *equipo;
};

struct NodoScuderia{
    struct Scuderia *datosScuderia;
    struct NodoScuderia *sig;
};

struct Formula1{
    struct NodoScuderia *headScuderia;
    struct NodoPiloto *headPilotos;
};
```

Se solicita que implemente la siguiente función:

```
struct Piloto *AsignarPilotoScuderia(struct Formula1 *f1, int idPilotoSaliente, int idPilotoEntrante)
```

Esta función recibe una referencia al sistema Formula 1 y dos id de pilotos (el saliente y el entrante). Esta función debe asignar un piloto en lugar de otro, es decir, reemplazar un piloto en la Scudería. El piloto para reemplazar es indicado por idPilotoSaliente y el reemplazo por el piloto identificado con idPilotoEntrante. Debe validar el caso en que alguno de los dos pilotos no esté en el sistema, por ende, no se podría llevar a cabo la operación. En caso de éxito retorna un puntero al piloto que fue reemplazado y en caso de fracaso retorna NULL.

Consideraciones obligatorias (si no se cumplen tiene nota mínima)
--

- Debe modularizar las soluciones e implementar lo solicitado.
- Comente los supuestos en caso de ser necesario.
- Respete las reglas y buenas prácticas de programación y desarrollo.