**UTN FRMDP**

**27/02/2023**

**Final Laboratorio II**

**IMPORTANTE:**

* **Realizar todas las funciones que se indican.**
* **Añadir su nombre y apellido al “main.c” y comentarios a su código identificando cada inciso.**
* **Si no se encuentra en Zoom durante el desarrollo del examen, la evaluación no será validada y por lo tanto su nota final será *Ausente.***

Enunciado

Un sistema de envíos aéreos y terrestres posee un registro de los destinos a los que se puede enviar un paquete dependiendo del origen de partida, ademas, el envio puede ser de tipo “aéreo” o “ferrocarril”, y tendrá su costo y tiempo que llevaría el envío a ese destino bajo ese tipo de logística.

Datos y desglose de la información:

| ***“registroEnvios.bin”*** | **nodoDestino** | **nodoOrigen** |
| --- | --- | --- |
| **typedef struct {  char origen[30];  char destino[30];**  **char tipo[30];*//”aereo”,“ferrocarril”*  float costo;**  **int tiempoViaje; }registroEnvios;** | **typedef struct nodoDestino {  char nombre[30];**  **float costo;**  **int tiempoViaje;**  **struct nodoDestino\* sig; }nodoDestino;** | **typedef struct nodoOrigen**  **{  char nombre[30];  nodoDestino\* destinosAereos;**  **nodoDestino\* destinosFerro;  struct nodoOrigen\* sig;**  **}nodoOrigen;** |

*Para aprobar el examen debe tener al menos un ejercicio recursivo correctamente resuelto*

| Obtenido | Valor | **Inciso** | |
| --- | --- | --- | --- |
|  | - | * Hacer una función main () que invoque a sus incisos y demuestre el correcto funcionamiento del programa. | |
|  | 25 | 1. **Cargar los datos desde el archivo**  * Leer cada uno de los registros del archivo y cargar una lista de orígenes donde posee el campo de la ciudad de origen y dos sublistas, una que posee todos los destinos que serán de tipo “aéreo” y otra que posee todos los destinos de tipo “ferrocarril”. Cada nodo de la sublista tendrá el nombre de la ciudad destino, el costo y el tiempo de viaje.   + **(Recursivo)** Cada ciudad origen se inserta o**rdenado por nombre** ascendente   + Cada ciudad destino se inserta **al final** de la lista (ambas sublistas) | |
|
|  | 15 | 1. Realizar una función **recursiva** que retorne el promedio de *tiempoViaje* de destinosAereos para un nombre de origen recibido por parámetro. | |
|  | 20 | 1. Realizar una función que persista en 2 archivos registros con la siguiente información:  | typedef struct {  int cantDestinosAereos;  char ciudadOrigen[30];  float costoPromedioAereo;  float tiempoPromedioAereo;  }destinosAereos; | typedef struct {  int cantDestinosFerro;;  char ciudadOrigen[30];  float costoPromedioFerro;  float tiempoPromedioFerro;  }destinosFerro; | | --- | --- | | |
|  | 15 | 1. Completar el TDA Pila otorgado adecuadamente con las funciones indicadas. | |
|  | 25 | 1. Realizar una función que elimine de la sublista los destinosAéreos que posean un tiempo de viaje menor a **3hs** y pase los **tiemposViaje** a una pila **AEREOS.**   **Si la sublista queda vacía, y la sublista de ferrocarriles también está vacía, se debe eliminar el nodoOrigen.** | |

**Tabla de puntuación:**

| **Obtenido** | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nota** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Condición** | **Desaprobado** | | | | | **Aprobado** | | | | |