
	Facultad de Ingeniería y Ciencias Agrarias	PROGRAMACIÓN ESTRUCTURADA
		Practica N° 05
	<i>Estructuras dinámicas y archivos binarios</i>	

Aclaración: Donde dice “función recursiva”, es OBLIGATORIO que la función sea RECURSIVA. En los ejercicios que dicen solamente “función” la recursividad es optativa.

- Ej. 1:** Realizar una función *recursiva* que imprima el contenido de una lista de enteros en forma inversa.
Si la lista contiene 14,17,20,23
el programa deberá imprimir: 23, 20, 17, 14
- Ej. 2:** Realizar una función que inserte un valor entero en una lista de enteros ordenada, preservando el orden.
- Ej. 3:** Hacer una función que elimina un nodo de una lista de enteros. La función recibirá como parámetros el puntero al primer elemento de la lista y el valor que contiene el nodo a ser eliminado. Si el valor estuviera repetido, se eliminará el primer nodo que lo contenga. Considere que la lista puede estar vacía o que el valor solicitado no se encuentre.
- Ej. 4:** Programar una función que elimina un nodo de una lista de enteros. La función recibirá como parámetros el puntero al primer nodo de la lista y la posición del nodo a eliminar. Considere que la lista puede estar vacía o que la posición del nodo a eliminar no exista.
- Ej. 5:** Programar las funciones `push` y `pop` de la implementación de pilas con listas simplemente enlazadas y las funciones `enqueue` y `dequeue` de la implementación de colas con listas simplemente enlazadas.
- Ej. 6:** Realizar cuatro funciones que impriman un árbol binario de enteros según el recorrido preorden, inorden, postorden y por niveles.
- Ej. 7:** Realizar una función que permita insertar un valor en un árbol binario de enteros de forma tal que su impresión en inorden imprima los números ordenados de menor a mayor. Considere que el árbol recibido puede estar vacío.
- Ej. 8:** Programar una función que permite eliminar un nodo de un árbol binario de enteros. La función recibirá como parámetros el puntero a la raíz del árbol y el entero que se desea eliminar. Se deberá eliminar el nodo que primero se encuentra (en búsqueda postorden) que contenga dicho entero y todos los nodos que dependen de éste. Considere que el árbol puede estar vacío o que el nodo a eliminar no exista.

	Facultad de Ingeniería y Ciencias Agrarias	PROGRAMACIÓN ESTRUCTURADA
		Practica N° 05
	<i>Estructuras dinámicas y archivos binarios</i>	

Ej. 9: Dada la siguiente declaración:

```
typedef struct {
    char dni[9];
    char nombre [100], apellido [100];
    double monto_adeudado;
    unsigned int dia, mes,anio;
}t_datos;
```

Realizar una función de carga que solicite al usuario que ingrese por teclado dichos datos y los grabe en el archivo binario “deudores.dat”, en modo append. La carga de datos finalizará cuando el usuario ingrese como DNI el cero. Considerar que habrá un registro por cada deuda de cada deudor por lo que habrá registros con igual DNI y distintos o iguales montos en caso que la persona hubiese contraído deudas similares en distintas fechas.

Ej. 10: Dado el struct del ejercicio 9, realizar una función que solicite al usuario un nombre del archivo y a continuación abra y muestre el contenido de ese archivo, en formato de listado por pantalla, separando los datos en columnas y mostrando en cada renglón un registro distinto.

Ej. 11: Dado el struct del ejercicio 9, realizar una función que reciba como parámetros los nombres de dos archivos. El primero contendrá los datos de las deudas, la función deberá abrir dicho archivo y traspasar los datos hacia el segundo archivo sin incluir registros con dni repetido. En caso de que haya repeticiones en el archivo original, deben sumarse todos los montos adeudados y grabarse en un único registro para cada DNI.