

Práctica de carácter obligatorio

Es momento de que apliques todo lo visto en esta sección, ya que de otro modo no podrás continuar con los temas que vienen a continuación. Si bien la propuesta de esta clase no tiene, ni por asomo, la dificultad de proyectos como *MasterMind* o *Buscaminas*, el hecho de enfrentarte a los punteros como nuevo tema te impondrá igualmente cierto reto que debes poder superar.

En esta práctica surgirán dudas y por tanto es el lugar para aprovechar al máximo a los docentes del curso. Pregunta, vuelve a preguntar, y pregunta nuevamente. Es así que mejor se aprende. Recuerda que *"no hay preguntas tontas sino tontos que no preguntan"*.

Lo que te propongo a continuación te permitirá aplicar lo siguiente:

- Creación de operaciones propias (*funciones y procedimientos*).
- Entendimiento profundo del modificador **var** y **const** en parámetros por referencia.
- Trabajo con **punteros** y memoria dinámica.
- Trabajo con **listas encadenadas simples**.

Si bien esta práctica no es de lo más interesante, es muy poderosa, y por tanto **OBLIGATORIA**.

Propuesta

Implementar en Pascal, con los conocimientos adquiridos hasta el momento y sin ninguna herramienta extra, las siguientes operaciones y tipos de datos.

1. Un tipo llamado **ListaNumeros** que represente a una lista encadenada de enteros.
2. Una función llamada **crearLista** que reciba como argumento un entero con el largo de la lista a crear, y retorne como resultado una lista encadenada de enteros del largo indicado (de tipo **ListaNumeros**). Cada nodo debe contener un entero al azar entre 0 y 9. Si el largo pasado como argumento es menor o igual que cero, se retorna una lista vacía (puntero **NIL**).
3. Un procedimiento llamado **imprimirLista** que reciba como argumento por referencia constante (**const**) la lista de tipo **ListaNumeros** e imprima en la salida estándar todos los valores que contiene, separados por espacio, en el orden en que aparecen en la lista.
4. Una función entera llamada **contarOcuurrencias** que reciba dos argumentos: un número entero y una lista de tipo **ListaNumeros** por referencia constante. La función retornará un entero indicando la cantidad de veces que el número pasado como argumento aparece en la lista. Si no aparece ninguna vez, obviamente se retornará cero (0).
5. Un procedimiento llamado **borrarLista** que reciba por referencia variable (**var**) un argumento de tipo **ListaNumeros** y elimine todos los nodos que contiene. El argumento debe quedar con el valor **NIL** al finalizar la operación.

6. Una función booleana llamada **borrarPrimeraOcurrencia** que reciba dos argumentos: un entero y, por referencia variable, una lista de tipo **ListaNumeros**. La función deberá recorrer la lista y borrar la primera aparición del número indicado. Si se encuentra el número y se hace una eliminación se retornará **TRUE**. Si el número indicado no estaba en la lista se retornará **FALSE**. Si el nodo eliminado era el único nodo de la lista ésta quedará vacía. Si la lista ya estaba vacía no se hará nada y se retornará **FALSE**.
7. Una función booleana llamada **borrarUltimaOcurrencia** que reciba dos argumentos: un entero y, por referencia variable, una lista de tipo **ListaNumeros**. La función deberá recorrer la lista y borrar la última aparición del número indicado. El resto del comportamiento es exactamente igual a **borrarPrimeraOcurrencia**.
8. Un procedimiento llamado **borrarOcurrencias**, el cual recibirá 4 argumentos:
 1. Un entero que se utilizará para buscar y eliminar de la lista.
 2. Una lista de tipo **ListaNumeros** por referencia variable.
 3. Un booleano que indicará si se deben eliminar solo las repeticiones del número o bien si se deben eliminar todas las apariciones del mismo.
 4. Un booleano que indicará si debe permanecer en la lista la primera ocurrencia del número, o bien, la última.

El comportamiento de esta operación es el siguiente: si se indica que se deben eliminar todas las apariciones del número pasado como argumento, entonces se borrarán todos los nodos en que dicho número aparece, sin dejar ninguno. En caso contrario, se tomará en cuenta el argumento booleano que indica si se ha de dejar la primera ocurrencia o la última. En el primer caso se eliminarán todos los nodos que contienen el número menos el primero en que aparece. En el segundo caso se eliminarán todos los nodos que contienen el número menos el último en que aparece.

La firma de estas funciones y procedimientos queda a tu criterio, siempre que cumplan con lo solicitado. Deben funcionar todas correctamente ya que para próximas clases las usaremos.

Entrega

Este ejercicio debe ser implementado en el programa principal, no en una unidad Pascal (**unit**). Deberás entregar tu archivo de código fuente, el cual será testeado por un docente y se te hará una devolución.

La entrega debe realizarse a bedelia@kaedusoft.edu.uy con el asunto: "**PRACTICA PUNTEROS UDEMY [NOMBRE] [APELLIDO] [DOCUMENTO]**" donde obviamente pondrás tu nombre, apellido y documentos reales.