**Practica de Laboratorio de Objetos y Abstracción de Datos**

**LISTAS ENLAZADAS**

Para empezar vamos a revisar algunos de los mecanismos básicos sobre cómo funcionan las listas enlazadas.

Imaginemos que tenemos una lista enlazada de números enteros. Definiremos una estructura Celda que representa una sola celda en la lista enlazada.

|  |
| --- |
| struct Celda {  int valor;  Celda\* next;  }; |

**Actividad 1:** Ahora empezaremos sumando los valores de todos los elementos de la lista enlazada. Escribimos la función **sumaDeElementosEn** como una versión iterativa:

|  |
| --- |
| /\* version Iterativa\*/  int sumaDeElementosEn(Celda\* lista) {  int resultado = 0;  for (Celda\* actual = lista; actual != NULL; actual = actual->next) {  resultado += actual->valor;  }  return resultado;  } |

Escribir una versión recursiva de: **int sumOfElementsIn(Celda\* lista);**

**Actividad 2:** En seguida tenemos una función que retorna el último elemento de una lista enlazada. Otra vez escribir una versión recursiva de esta función.

|  |
| --- |
| Celda\* ultimoElementoDe(Celda\* lista) {  if (lista == NULL) cout << "Lista vacia no tiene ultimo elemento." << endl;  /\* Itera hacia el frente hasta que el actual puntero next de celda sea null. Es ahí  \* donde la lista finaliza.  \*/  Celda\* resultado = lista;  while (resultado->next != NULL) {  resultado = resultado->next;  }  return resultado;  } |

**Actividad 3:** Escribir una función

Celda\* concatena(Celda\* uno, Celda\* dos);

que toma como entrada dos listas enlazadas, luego concatena la segunda lista en la parte posterior de la primera lista enlazada. Su función debe retornar un puntero al primer elemento de la lista resultante.

A continuación, actualizar la función para que tenga esta firma:

void concatena(Celda\*& uno, Celda\* dos);

Actividad 4: Escribir una función recursiva para imprimir una lista enlazada en reversa.

void ImprimeReversa(Celda\* lista);