
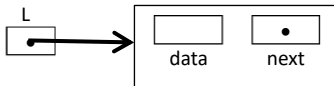


## CIS 2101 Practice Activity

A linked list can be implemented in computer's memory WITH or WITHOUT header. A header cell is a dynamically allocated node or cell that will not store any data. It is used to have uniformity in accessing and hence making the code concise but with an additional storage overhead.

### Illustration of an initially empty list:

Without Header cell	With Header Cell
	
Definition of a Linked list	<pre>typedef struct node {     int data;     struct node *next; }*List;</pre>

Given the definition of data type List, write the code of the following functions:

#### A] Initialize the list to be empty [With header cell vs. Without Header Cell]

- 1) `initListWithHeader()` – The function will initialize the given list to be empty by letting the given list point to a dynamically allocated header cell. See illustration above.
- 2) `initListNoHeader()` – The function will initialize the given list to be empty by simply making it NULL. See illustration above.

#### B] Function `insertLast()`. The function will insert a given element at the last position of the given list.

Write the code of function `insertLast()` using the following versions:

- 1) List with header cell: This list variation will use a look-ahead technique for traversal.
- 2) List without header cell:
  - a) Version 1: Using a Pointer to Pointer to Node (PPN) to traverse
  - b) Version 2: Using a look-ahead strategy to traverse.

#### C] Function `deleteElem()`. The function will delete the first occurrence of the given element from the list.

Write the code function `deleteElem()` using the following versions:

- 1) List without header cell using PPN.
- 2) List without header cell using look-ahead technique