**Purpose**

* To design a template linked list class.

**Requirements**

* Support a function that checks if a certain value is present in the list.
* Support a function to pop elements to the front
* Support a function to insert elements at arbitrary indices

**Classes**

* Node:
  1. Stores the data element of the given type, as well as the link to the next node.
  2. Data:
     1. data of any type
     2. pointer to another node.
* LinkedList:
  1. Manages the first node and the overall list
  2. Data
     1. A pointer to the head of the list
     2. The length of the list

**High level architecture:**

* To check if a value is in the list:
  1. Set a pointer Node ptr to the head of the list.
  2. If ptr is null, return false.
  3. If the data in ptr is the value, return true.
  4. Otherwise, make ptr point to the next field of ptr.
  5. Go back to step 2.
* To remove the first element of the list:
  1. If the list is empty, the operation is invalid.
  2. Create a temperory Node temp.
  3. Point temp to the next field of the head of the list.
  4. Delete the node pointed to by the head.
  5. Make node point to temp.
  6. Update the length of the list.
* To insert an element to position ‘index’:
  1. If the index is greater than the length, the operation is invalid.
  2. Create a pointer to the new node, called newNode, a pointer to the head called ptr, and a temporary pointer temp.
  3. If the index is zero:
     1. Point temp to the head.
     2. Point the head to newNode
     3. Point the head’s next to temp.
  4. Otherwise:
     1. Make ptr point to the element at index-1.
     2. Point temp to ptr’s next.
     3. Point newNode’s next to temp.
  5. Update the length of the list

**User Interface:**

* The program uses a text-based interface over the command line.
* The program does not support input, it uses data hard-coded into the source.

**Test Cases:**

* Tested the contains() by looking for an element present in the list and an element absent from the list.
* Tested the pop method by repeatedly popping until the list is empty, and then trying to pop again
* Tested the push method by pushing to invalid indices, to the start of the list, the end of the list, and at random valid indices.