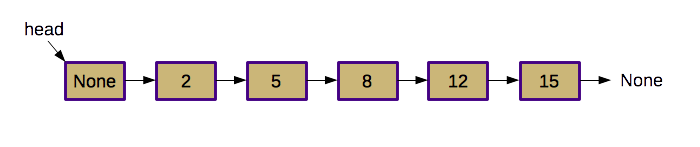
1. Write a code for splay tree insertion ( Bottom up ) . show the insertion by inserting 20, 10, 30, 40, 45, 56, 70 into an empty splay.
2. Write a code for splay tree deletion ( top down) . show the deletion by deleting element 45 from above tree.
3. Below is the linked list representation,



1. Create a skip list representation of above linked list by writing a code and show the result ( ATTACH THE SCREENSHOT OF YOUR RESULT ) . Here is the pseudocode for insertion :

*Check whether the new value is already in the list; abort if found*

*Flip coins to determine the insertion height.*

*If necessary, increase sentinel height until it is one greater*

*than the insertion height.*

*Initialize* cur *to be the head of the list.*

*While you haven't run off the bottom of the list:*

*Find the possible insertion point at this level by advancing* cur

*until* cur.next *is* None*, or the* cur.next.value *is*

*greater than than the value to add.*

*If insertion should happen at this level:*

*Insert a new node after* cur*.*

*Update the below reference of the most recently added node.*

*Move* cur *to the next lower level.*

* You can promote any node of your liking to the next level.

1. Write a code for searching element in skip list . Use that code for search value 14 and 12.

(attach the screenshot of your result search) .

1. Write a code to find the longest common prefix using the trie data structure . Apply that code to find the longest common prefix of

Cancellation, cancel, Canada, canabary , can, cando.

* Lexicographic sorting of a set of keys can be accomplished with a simple [**Trie-based algorithm**](https://www.techiedelight.com/trie-implementation-insert-search-delete/) as follows:
* Insert all keys into a Trie.
* Print all keys in the Trie by performing [preorder traversal](https://www.techiedelight.com/preorder-tree-traversal-iterative-recursive/) on Trie to get output in lexicographically increasing order.

1. Write a code for lexicographic sorting of keys using trie data structure :

Your code must ask for few words to be entered and result must be showing the lexicographically sorted words.

EXAMPLE : banana, aeroplane, basket, plastic.

Result : aeroplane, banana, basket, plastic.