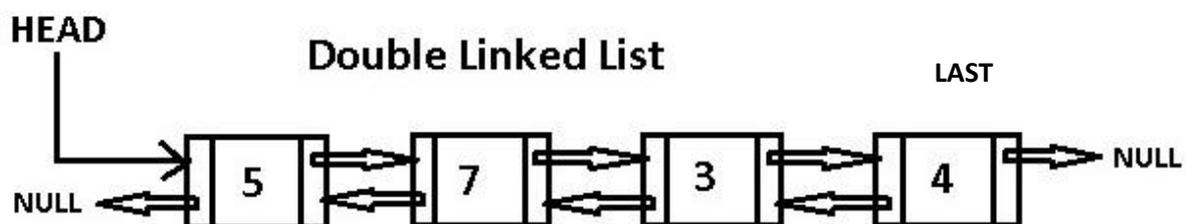


# Doubly Linked Lists

**NB.** Unless they say otherwise an exam question referring to a linked list will mean a singly linked list.

There needs to be a way to get around the, traverse one way only, restriction of a single linked list.

So let's make each node have 2 pointers - one forward and one back. We also have the data. We also have pointers to the head or First and Last nodes.



The additional pointer for each element makes it possible to traverse the list in either direction and from any element.

Insertion and deletion are now much easier as we have instantly the next and previous pointers.

It is even possible to **delete** elements with a single pointer variable because you can use the previous pointers to delete elements behind your traversal current position.

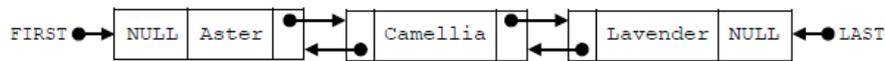
The **downside** to double linked lists is now we have doubled the space needed in memory.

## Practical use

Say you want to create a music player. A double linked list would nicely implement: play, next track, previous track, delete a track and insert a new track etc really well. Web browser forward and back buttons. Undo and redo in applications.

## Inserting a new node

Consider the following doubly linked list which holds the names of flowers in alphabetical order.

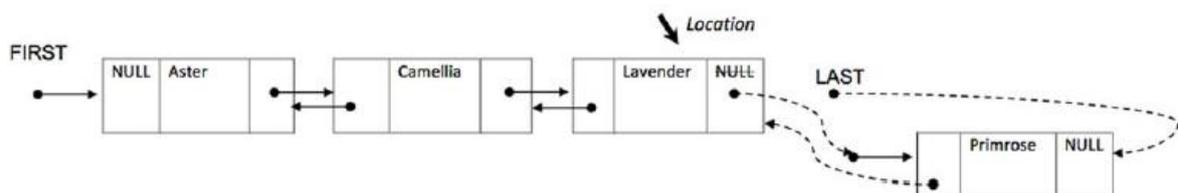


- (b) Explain how "Primrose" could be inserted into this doubly linked list. You should draw a labelled diagram in your answer.

[6]

The key here for the exam is to draw the finished linked list **and** then explain it.

This will help you visualise the steps better.



1. We created a new node and instantiated it with a value Primrose and 2 for now null previous and next pointers.
2. We started at position FIRST and performed a sequential search
3. Compared nodes as we went until we found the correct position which is after Lavender
4. The primrose next field will point to NULL and the previous to Lavender
5. We also needed to change Lavenders next NULL pointer to now point to Primrose
6. Reset the LAST pointer to be the new node

