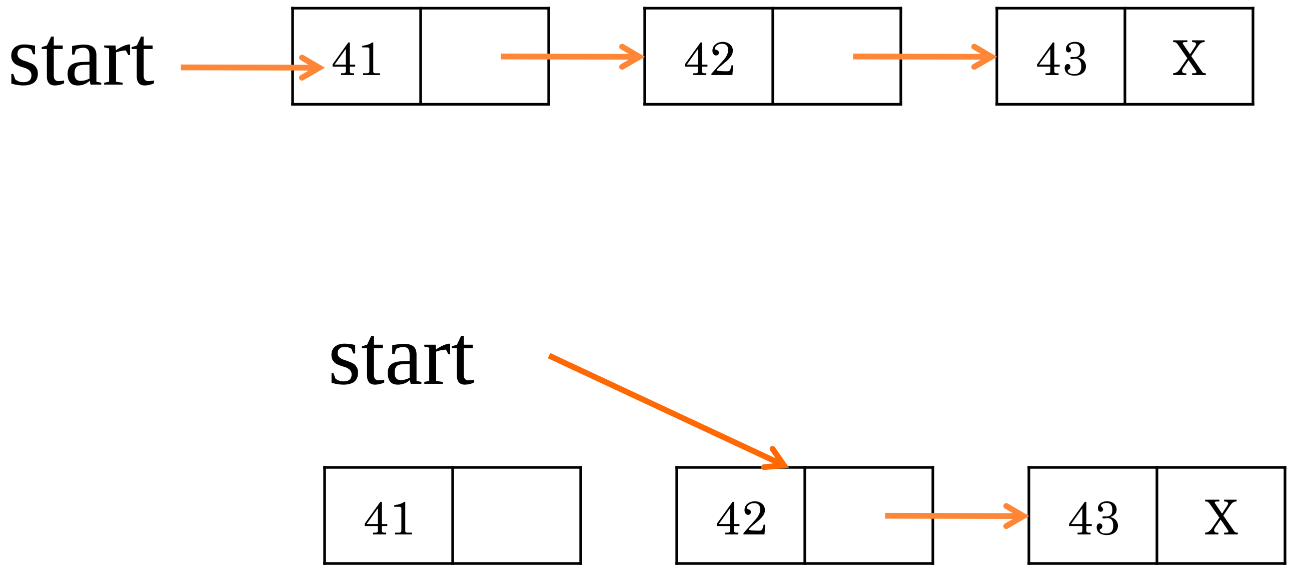
## Deletion

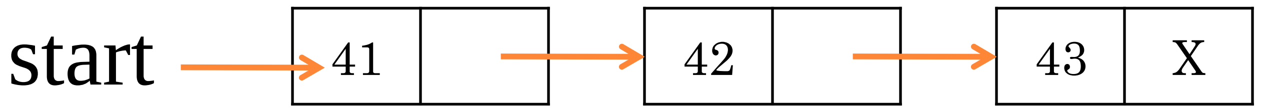
❑Delete a node from the linked list, we have three ways:

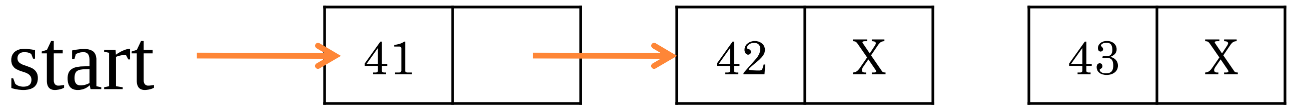
i. At beginning
ii. At end
iii. At middle

## Deletion at beginning(del)

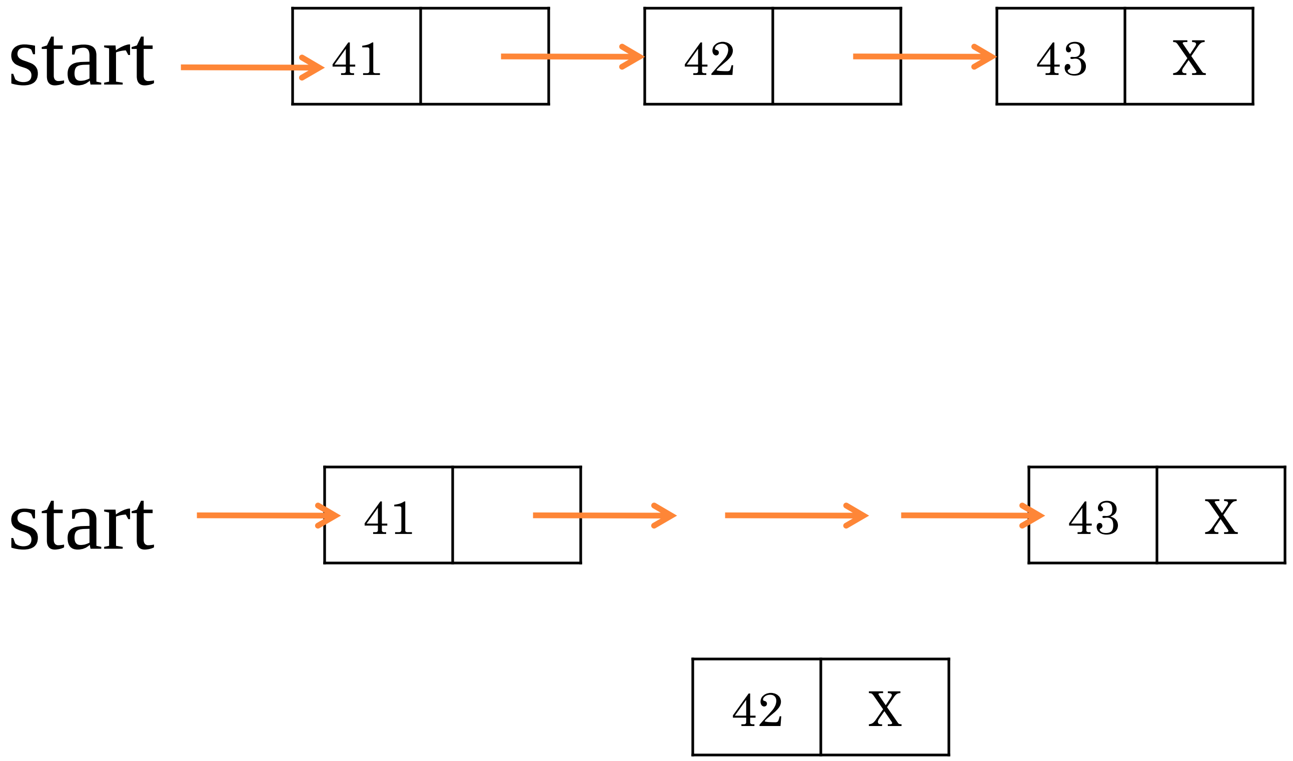


## Deletion at End(delendlink)





# Deletion at Middle(delmidlink)



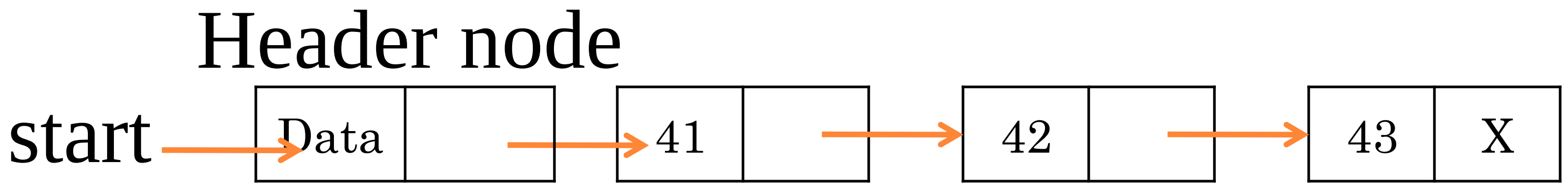
## Searching in a linked list

- ❑We have two types of linked list, sorted and unsorted linked lists.
- ❑Searching an item in an unsorted list involves traversing through the list and comparing the item with the data field of each node, one by one, of linked list.
- ❑The search may either be terminated upon finding the first match or may continued till the end of the list.
- ❑Unsorted(searchlink).

## Header linked list

❑A header linked list is a list which always contains a special node, called header node, at the beginning of the list.

❑The information field of the header node may be kept unused or it could be used to hoed global information about the header.



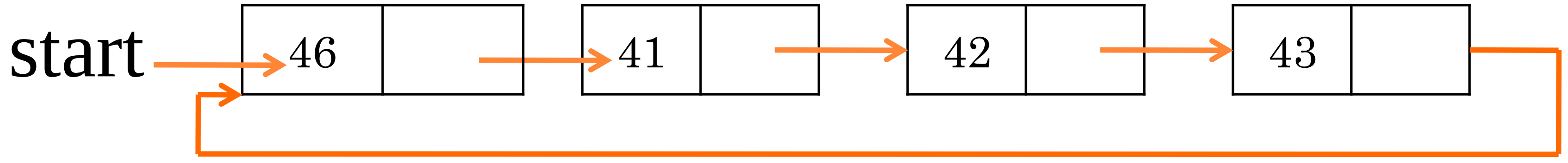
❑The first node in header list is the header list is the node following the header node and the location of the first node, not start, as with ordinary linked list.

## Header linked list

- Prgm: creation of a header linked list:
- The header data is used to store node count.
- (Header)

## Circular linked list

- ❑A linked list whose last node points back to the first node instead of containing the null pointer is called a circular list.
- ❑The advantage of the circular list is that every node has a predecessor.
- ❑The advantage of this property is that given a pointer to a node, we can reach at any node in the list.
- ❑A circular list does not have a first or last node.
- ❑We must, therefore, establish a first or a last node or both by convention.



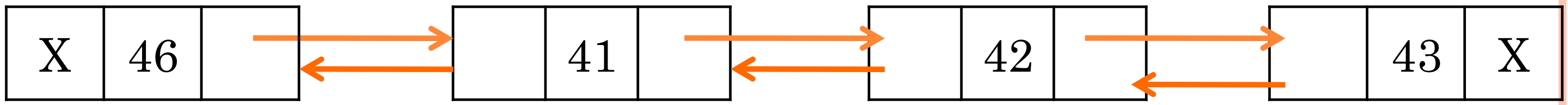
## Two - way linked list

- Each list discussed so far is called a one – way list, since there is only one way the list can be traversed.
- That is, beginning with the list pointer variable start, which points to  the node, and using the next pointer field next to point to the next node in the list, we can traverse the list in only one direction.
- In such a list, given the location of  node, one has immediate access to the next node in the list, but one does not have access to the proceeding node without traversing the part of the list.
- A two – way list or **doubly – linked list** is a list that can be traversed in two directions: in the usual forward direction from the beginning of the list to the end, or in the location direction from the end of the list, one now has immediate access to both the next node and the proceeding node in the list.

## Two - way linked list

❑A two – way list can be defined as a linear collection of data elements, called nodes, where each node consists of the following three parts:

i. An information field which contains the data
ii. A next pointer field which contains the location of the next node in the list
iii. A previous pointer field that contains of the preceding node in the list.


X
46
41
42
43
X