

## Section B: Solving the programming tasks

This section contains 3 tasks. Total score for this section is 12(4×3)

**Note:** You should cover all the cases including the case of empty list.

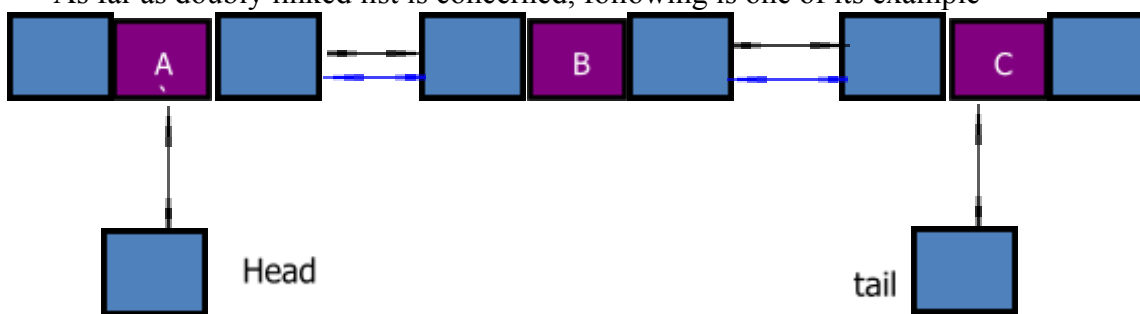
1. Create a function named `insertAtFront` with absence of return type that store the data in a new node, and insert that node at the front of a singly linked list.

2. A doubly linked list has the following structure for the node

```
struct DoublyNode
{
    int data;
    DoublyNode *next, *prev;
};
```

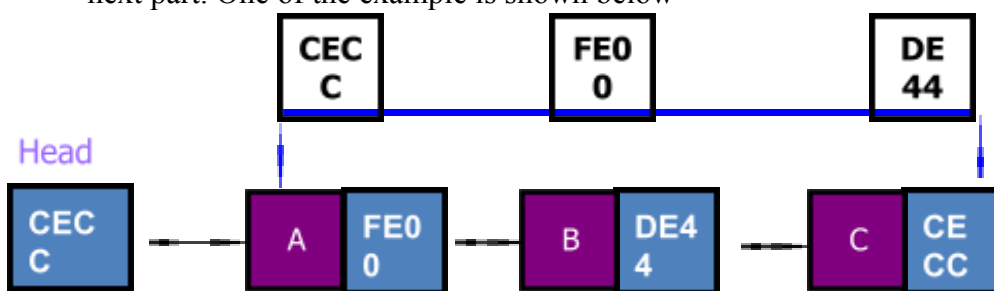


As far as doubly linked list is concerned, following is one of its example



You are required to create a function named `DoublyFrontInsertion` that always insert the data as the first node of doubly linked list.

3. The circular singly linked list has the same Node structure as of Singly linked list. However, for the List structure, the last node is pointing towards the first node instead of having NULL stored in its next part. One of the example is shown below

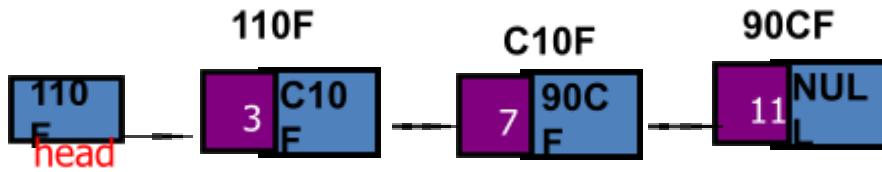


Create a function named `CircularFrontInsertion` that always insert the data as first node of the circular linked list.

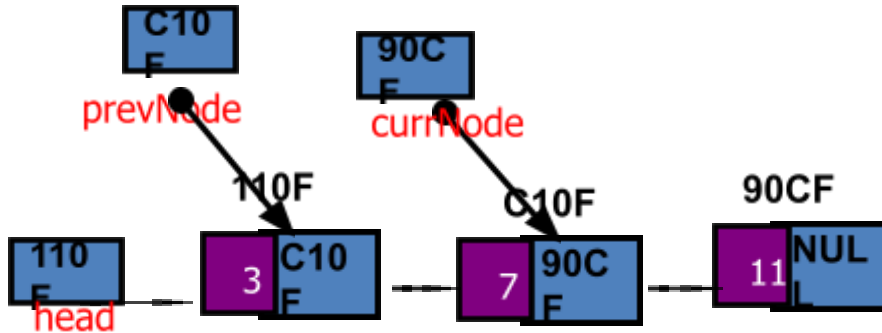
## Section C: Error correction

This section consist of only one question. Total score for this section is 3.

1. Suppose that someone wants to create a function named `deleteFromMiddle`. Suppose we have the linked list given below.



We want to delete the node that has the value of 7 stored in it. After some pre-processing and using two extra pointers, we have the following situation



In order to delete the node, someone try the following instructions for the deletion

```
delete current;
prevNode->next = prevNode->next->next;
```

Can you figure out

- The problem in this approach?
- How can you solve this problem?