Reverse k Nodes:

```
Node *reversek(Node *&head, int k)
{

    // take 3 pointers to reverse the first k nodes using iterator method
    Node *prev = NULL;
    Node *curr = head;
    Node *next;
    // take a count variable to know where to stop iterating
    int count = 0;
    while (curr != NULL && count < k)
    {
        next = curr->next;
        curr->next = prev;
        prev = curr;
        curr = next;
        count++;
    }
    // call recursively for the rest of the linkedlist
    // previous is pointing to the last node of the first k nodes and next is
pointing to k+1th node
    //  we have 2->1 and 4->3->6->5 we need to point 2->4
    if (next != NULL)
    {
        head->next = reversek(next, k);
    }
    return prev;
}
int main()
{
    Node *head = NULL;
    insertAtTail(head, 3);
    insertAtTail(head, 36);
    insertAtTail(head, 2);
    insertAtTail(head, 10);
    insertAtTail(head, 9);
    insertAtHead(head, 111);
    insertAtHead(head, 777);
    insertAtHead(head, 26);
    cout << "New LinkedList" << endl;
    display(head);
    int k=2;
    Node* newhead3=reversek(head,k);
    display(newhead3);
    return 0;
}
```

```
Output: 26->777->111->3->36->2->10->9->NULL
        777->26->3->111->2->36->9->10->NULL
```