

## Swap Kth node from beginning with Kth node from end in a Linked List

Given a singly linked list, swap kth node from beginning with kth node from end. **Swapping of data is not allowed, only pointers should be changed.** This requirement may be logical in many situations where the linked list data part is huge (For example student details line Name, RollNo, Address, ..etc). The pointers are always fixed (4 bytes for most of the compilers).



For  $k = 3$ , the above list should be changed to following (6 and 3 are swapped)



The problem seems simple at first look, but it has many interesting cases.

Let X be the kth node from beginning and Y be the kth node from end. Following are the interesting cases that must be handled.

- 1) Y is next to X
- 2) X is next to Y
- 3) X and Y are same
- 4) X and Y don't exist (k is more than number of nodes in linked list)

We strongly recommend you to try it yourself first, then see the below solution. It will be a good exercise of pointers.

```
// A C++ program to swap Kth node from beginning with kth node from end
#include <iostream>
#include <stdlib.h>
using namespace std;

// A Linked List node
struct node
{
```

```

    int data;
    struct node *next;
};

/* Utility function to insert a node at the beginning */
void push(struct node **head_ref, int new_data)
{
    struct node *new_node = (struct node *) malloc(sizeof(struct node));
    new_node->data = new_data;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}

/* Utility function for displaying linked list */
void printList(struct node *node)
{
    while (node != NULL)
    {
        cout << node->data << " ";
        node = node->next;
    }
    cout << endl;
}

/* Utility function for calculating length of linked list */
int countNodes(struct node *s)
{
    int count = 0;
    while (s != NULL)
    {
        count++;
        s = s->next;
    }
    return count;
}

/* Function for swapping kth nodes from both ends of linked list */
void swapKth(struct node **head_ref, int k)
{
    // Count nodes in linked list
    int n = countNodes(*head_ref);

    // Check if k is valid
    if (n < k) return;

    // If x (kth node from start) and y(kth node from end) are same
    if (2*k - 1 == n) return;

    // Find the kth node from beginning of linked list. We also find
    // previous of kth node because we need to update next pointer of
    // the previous.
    node *x = *head_ref;
    node *x_prev = NULL;
    for (int i = 1; i < k; i++)
    {
        x_prev = x;
        x = x->next;
    }

    // Similarly, find the kth node from end and its previous. kth node
    // from end is (n-k+1)th node from beginning
    node *y = *head_ref;
    node *y_prev = NULL;
    for (int i = 1; i < n-k+1; i++)
    {

```

```

        y_prev = y;
        y = y->next;
    }

    // If x_prev exists, then new next of it will be y. Consider the case
    // when y->next is x, in this case, x_prev and y are same. So the statement
    // "x_prev->next = y" creates a self loop. This self loop will be broken
    // when we change y->next.
    if (x_prev)
        x_prev->next = y;

    // Same thing applies to y_prev
    if (y_prev)
        y_prev->next = x;

    // Swap next pointers of x and y. These statements also break self
    // loop if x->next is y or y->next is x
    node *temp = x->next;
    x->next = y->next;
    y->next = temp;

    // Change head pointers when k is 1 or n
    if (k == 1)
        *head_ref = y;
    if (k == n)
        *head_ref = x;
}

// Driver program to test above functions
int main()
{
    // Let us create the following linked list for testing
    // 1->2->3->4->5->6->7->8
    struct node *head = NULL;
    for (int i = 8; i >= 1; i--)
        push(&head, i);

    cout << "Original Linked List: ";
    printList(head);

    for (int k = 1; k < 10; k++)
    {
        swapKth(&head, k);
        cout << "\nModified List for k = " << k << endl;
        printList(head);
    }

    return 0;
}

```

[Run on IDE](#)

Output:

Original Linked List: 1 2 3 4 5 6 7 8

Modified List for k = 1

8 2 3 4 5 6 7 1

Modified List for k = 2

8 7 3 4 5 6 2 1

Modified List for k = 3

8 7 6 4 5 3 2 1

Modified List for k = 4

8 7 6 5 4 3 2 1

Modified List for k = 5

8 7 6 4 5 3 2 1

Modified List for k = 6

8 7 3 4 5 6 2 1

Modified List for k = 7

8 2 3 4 5 6 7 1

Modified List for k = 8

1 2 3 4 5 6 7 8

Modified List for k = 9

1 2 3 4 5 6 7 8

Please note that the above code runs three separate loops to count nodes, find x and x prev, and to find y and y\_prev. These three things can be done in a single loop. The code uses three loops to keep things simple and readable.

Thanks to [Chandra Prakash](#) for initial solution. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



The advertisement features a man in a white shirt pointing at a computer screen displaying code. The text 'Learn to Program in Python' is prominently displayed in large white letters. Below it, 'Python for Data Science' and 'A free online course from Microsoft' are written in smaller white text. At the bottom left is the EdX logo with the URL 'www.edx.org'. At the bottom right is a green button with the text 'Enroll Now'.

91 Comments Category: [Linked Lists](#)

## Related Posts:

- Merge two sorted linked lists such that merged list is in reverse order
- Compare two strings represented as linked lists
- Rearrange a given linked list in-place.
- Sort a linked list that is sorted alternating ascending and descending orders?
- Select a Random Node from a Singly Linked List
- Merge Sort for Doubly Linked List
- Point to next higher value node in a linked list with an arbitrary pointer
- Swap nodes in a linked list without swapping data

(Login to Rate and Mark)

3

Average Difficulty : **3/5.0**  
Based on **1** vote(s)

☐

Add to TODO List

☐

Mark as DONE

Like Share 36 people like this.

Writing code in comment? Please use [code.geeksforgeeks.org](http://code.geeksforgeeks.org), generate link and share the link here.

@geeksforgeeks, Some rights reserved

[Contact Us!](#)

[About Us!](#)

[Advertise with us!](#)