

GeeksforGeeks

A computer science portal for geeks

[Android App](#) [GeeksQuiz](#)

[Login/Register](#)

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Ask a Q](#)
- [About](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFacts](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

[String](#)

[Tree](#)

[Graph](#)

Move last element to front of a given Linked List

Write a C function that moves last element to front in a given Singly Linked List. For example, if the given Linked List is 1->2->3->4->5, then the function should change the list to 5->1->2->3->4.

Algorithm:

Traverse the list till last node. Use two pointers: one to store the address of last node and other for address of second last node. After the end of loop do following operations.

- Make second last as last (secLast->next = NULL).
- Set next of last as head (last->next = *head_ref).
- Make last as head (*head_ref = last)

```
/* Program to move last element to front in a given linked list */
#include<stdio.h>
#include<stdlib.h>

/* A linked list node */
struct node
{
    int data;
    struct node *next;
};

/* We are using a double pointer head_ref here because we change
   head of the linked list inside this function.*/
void moveToFront(struct node **head_ref)
{
    /* If linked list is empty, or it contains only one node,
       then nothing needs to be done, simply return */
    if(*head_ref == NULL || (*head_ref)->next == NULL)
        return;

    /* Initialize second last and last pointers */
    struct node *secLast = NULL;
    struct node *last = *head_ref;

    /*After this loop secLast contains address of second last
    node and last contains address of last node in Linked List */
    while(last->next != NULL)
    {
        secLast = last;
        last = last->next;
    }

    /* Set the next of second last as NULL */
    secLast->next = NULL;

    /* Set next of last as head node */
    last->next = *head_ref;

    /* Change the head pointer to point to last node now */
    *head_ref = last;
}

/* UTILITY FUNCTIONS */
/* Function to add a node at the beginning of Linked List */
void push(struct node** head_ref, int new_data)
{
    /* allocate node */
    struct node* new_node =
        (struct node*) malloc(sizeof(struct node));

    /* put in the data */
    new_node->data = new_data;
```

```

/* link the old list off the new node */
new_node->next = (*head_ref);

/* move the head to point to the new node */
(*head_ref) = new_node;
}

/* Function to print nodes in a given linked list */
void printList(struct node *node)
{
    while(node != NULL)
    {
        printf("%d ", node->data);
        node = node->next;
    }
}

/* Driver program to test above function */
int main()
{
    struct node *start = NULL;

    /* The constructed linked list is:
    1->2->3->4->5 */
    push(&start, 5);
    push(&start, 4);
    push(&start, 3);
    push(&start, 2);
    push(&start, 1);

    printf("\n Linked list before moving last to front ");
    printList(start);

    moveToFront(&start);

    printf("\n Linked list after removing last to front ");
    printList(start);

    getchar();
}

```

Time Complexity: $O(n)$ where n is the number of nodes in the given Linked List.

Please write comments if you find any bug in above code/algorithm, or find other ways to solve the same problem.

Related Topics:

- [Clone a linked list with next and random pointer | Set 2](#)
- [Given a linked list of line segments, remove middle points](#)

- [Construct a Maximum Sum Linked List out of two Sorted Linked Lists having some Common nodes](#)
- [Given a linked list, reverse alternate nodes and append at the end](#)
- [Pairwise swap elements of a given linked list by changing links](#)
- [Self Organizing List | Set 1 \(Introduction\)](#)
- [Merge a linked list into another linked list at alternate positions](#)
- [QuickSort on Singly Linked List](#)

Like

{ 0 }

Tweet

{ 0 }

g+1

{ 0 }

Writing code in comment? Please use ideone.com and share the link here.

33 Comments

GeeksforGeeks

 Login ▼ Recommend Share

Sort by Newest ▼



Join the discussion...

**Ekluv** • 7 days ago

My code

```
void movelast2first(node **head){  
  
    node* t=*head;  
  
    node* y;  
  
    while(t->next->next!=NULL){  
  
        t=t->next;  
  
    }  
  
    y=t->next;  
  
    t->next=NULL;  
  
    y->next=*head;  
  
    *head=y;  
  
}
```

^ | v • Reply • Share ›

**adasd** • 7 days ago

adadas

^ | v • Reply • Share ›

**adevakar verma** • 8 days ago

we can also swap data of first node and last node.

^ | v • Reply • Share ›

**Anshuman Singh** • 20 days ago

```
struct node * move(struct node * start)
{
    struct node *p,*last;
    p=start;
    while(p->next->next!=NULL)
    p=p->next;
    last=p->next;
    last->next=start;
    start=last;
    p->next=NULL;
    return start;
}
```

^ | v • Reply • Share ›

**Utkarsh Mishra** • a month ago

```
node* move(node* head)
{
    node *loop=head;
    while(loop->link->link!=NULL)
    loop=loop->link;
    loop->link->link=head; //make circular
    head=loop->link; //change head
    loop->link=NULL; //break circle
    return head;
}
```

^ | v • Reply • Share ›

**Hinata Hyuga** • 2 months ago

we can do it with single node.

1.traverse through the list till second last.

2.use temp pointer to store the last. and make second last->next = NULL.

3.temp->next = *head

4.*head = temp.

^ | v • Reply • Share ›

**Guest** • 2 months ago

```
int getcount(struct node *head)
{
    struct node *curr=head;
```

```
int count=0;
while(curr!=NULL &&curr->next!=NULL)
{
    count++;
    curr=curr->next;
}
return count;
}
```

```
void movelast(struct node *head)
{
    struct node *curr=head;
    struct node *p=head;
    struct node *q=head;
    int n=getcount(head);
```

[see more](#)[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Naval** • 3 months ago

using only one pointer no need to use 2 pointer last and second last we can done it using only second last pointer

here is the code

```
#include<stdio.h>
#include<stdlib.h>
```

```
struct node
{
    int data;
    struct node* next;
};
```

```
void moveFront(struct node **);
```

```
void push(struct node** head_ref, int new_data)
{
    struct node* new_node =(struct node*) malloc(sizeof(struct node));
```

[see more](#)[^](#) | [v](#) • [Reply](#) • [Share](#) ›**RajaCEGian** ➔ Naval • 3 months ago

Thank you

^ | v • Reply • Share ›



vamshi reddy • 3 months ago

Algorithms:

1. traverse the whole list till last node.
- 2.swap the first node data with last node data.

code:

<http://coliru.stacked-crooked....>

^ | v • Reply • Share ›



happysshao • 5 months ago

I think the sample code is not correct.

1->2->3->4;

will output 4->1->2->3

^ | v • Reply • Share ›



neeraj kumar → happysshao • 4 months ago

This is what needed!

^ | v • Reply • Share ›



Kim Jong-il • 7 months ago

Simple Code

```
struct node *MoveLastElementFirst(struct node *head)
{
    struct node *next,*prev;

    if(head==null || head->link == NULL)
        return head;

    prev=head;
    next = head->link;
    while(next->link!=NULL)
    {
        prev=next;
        next = next->link;
    }

    prev->link=NULL;
    next->link=head;
    head = next;
```

```
return head;
```

```
}
```

^ | v • Reply • Share ›



Vivek • 8 months ago

```
head=reverse(head)
```

```
head->next=reverse(head->next)
```

i think this should do

1 ^ | v • Reply • Share ›



Kim Jong-il → Vivek • 7 months ago

I do not think it work first, if it works then its inefficient.

1 ^ | v • Reply • Share ›



Jaiwardhan Swarnakar • 8 months ago

Go to the last node, link it with the main root node and return back recursively

here is the ideone link

<http://ideone.com/qbfxsq>

1 ^ | v • Reply • Share ›



ALEX • 10 months ago

```
/*stop at second last node*/
```

```
while(secondLastNode->next->next)
```

```
secondLastNode=secondLastNode->next;
```

```
/*make linked list circular*/
```

```
secondLastNode->next->next=head;
```

```
head=secondLastNode;
```

```
/*put a NULL */
```

```
secondLastNode->next=NULL;
```

^ | v • Reply • Share ›



wishall • 10 months ago

Bug:head node link should be made pointing 2 NULL,,,,

```
(*head_ref)->next=NULL; before *head_ref=last;
```

^ | v • Reply • Share ›



popeye → wishall • 10 months ago

No. We are not swapping head & the last node.

^ | v • Reply • Share ›



wishall → popeye • 10 months ago

yes,i was wrong,,,no bug

^ | v • Reply • Share ›

**Akash Panda** · a year ago

```
void MoveLastToFront(struct node **head)
{
    struct node *current=*head;

    if(current==NULL || current->next==NULL)
        return;

    while(current->next->next!=NULL)
    {
        current=current->next;
    }

    struct node *temp=current->next;

    current->next=NULL;

    temp->next=*head;

    *head=temp;
}
```

^ | v · Reply · Share ›

**Himanshu Dagar** · a year ago

even we can do it with a single pointer by keep track of forward nodes frm current node

^ | v · Reply · Share ›

**Guest** · a year ago

```
void move_last_node_to_beg(struct node **head)
{
    struct node **temp=&((*head)->link); //temp holds address of link part of 1st node which is
    pointed to by head node
    if(*temp!=NULL) //this is just to handle the case that the 1ST node itself is not the last node
    { while((*temp)->link)!=NULL) //find the address present in the link field of 1st node by
    *temp, then go to that address and check if that node's link field is null then quit the loop
    , temp=&((*temp)->link); //this is basically to make temp to hold the next node's link field's
    address
    //finally temp will hold address of the last but 1 node's link field's address...bcoz while loop
    quits when the next node's link field contains null
}
```

```
(*temp)->link=*head; //now change the address of the present in the last nodes link to make it
point to the head node
*head=*temp; //head node now points to where earlier the last but 1 node's link field was
pointing that is to the last node
*temp=NULL; //the last but 1 node's link field now contains null
}
}
```

^ | v • Reply • Share ›



adithya • 3 years ago

```
/* Make last node first */
void reverse(node **head) {
    node *temp,*temp1;
    temp=*head;
    temp1=*head;
    temp=temp->link;
    while(temp1->link!=NULL) {
        temp=temp->link;
        temp1=temp1->link->link;
    }
    temp1->link=*head;
    *head=temp1;
    temp->link=NULL;
    return;
}
```

^ | v • Reply • Share ›



adithya • 3 years ago

```
/* Function for making lastnode first*/
void reverse(node **head) {
    node *temp,*temp1;
    temp=*head;
    temp1=*head;
    temp=temp->link;
    while(temp1->link!=NULL) {
        temp=temp->link;
        temp1=temp1->link->link;
    }
    temp1->link=*head;
    *head=temp1;
    temp->link=NULL;
    return;
}
```

^ | v • Reply • Share ›



Venki • 4 years ago

Function to move last node to start of the list with only one crawl pointer. Comments explains the logic.

```
void moveToFront(struct node **head_ref)
{
    /* Proceed only when list is valid (efficient code) */
    if( *head_ref && (*head_ref)->next )
    {
        struct node *ite = *head_ref;

        /* Move to second last node */
        while( ite && ite->next && ite->next->next )
        {
            ite = ite->next;
        }

        /* Make the list circular */
        ite->next->next = *head_ref;
        /* Set up new head */
        *head_ref = ite->next;
        /* Break the loop */
        ite->next= NULL;
    }
}
```

3 ^ | v • Reply • Share ›



ashish jaiswal → Venki • 3 months ago

venki...thats good...really...

^ | v • Reply • Share ›



Murali S Iyengar → Venki • a year ago

@Venki

The check "ite && ite->next" in the while loop is redundant as you have already checked for head and head->next in "if" at the beginning.

The while loop may be changed to

```
while (ite->next->next)
{
```

```
ite = ite->next;
```

```
}
```

1 ^ | v • Reply • Share ›



renu → Venki • a year ago

awesome!!!

^ | v • Reply • Share ›



Coder → Venki • 2 years ago

Nice approach really good Venki :)

^ | v • Reply • Share ›



Soumya Sengupta → Venki • 2 years ago

@venki-great iterative code.....enjoyed it...

```
/* Paste your code here (You may delete these lines if not writing code) */
```

^ | v • Reply • Share ›



Sambasiva • 5 years ago

```
void moveToFront(struct node **head_ref)
{
    struct node *p = *head_ref;
    if(!p || !p->next) return;
    for(;p->next->next; p = p->next);
    p->next->next = *head_ref;
    *head_ref = p->next;
    p->next = NULL;
}
```

^ | v • Reply • Share ›



Sam • 5 years ago

Below is C# version

```
public static LinkedList MoveLastItemToFirst(LinkedList head)
{
    LinkedList last = null;
    LinkedList secondLast = null;
    LinkedList cur = head;

    while (null != cur)
    {
```

```
        secondLast = last;
        last = cur;
        cur = cur.Next;
    }

    if (null != last)
    {
        secondLast.Next = null;
        last.Next = head;
        head = last;
    }

    return head;
}
```

^ | v • Reply • Share ›

✉ Subscribe

D Add Disqus to your site

🔒 Privacy



GeeksforGeeks

Like

93,326 people like GeeksforGeeks.



Feedback social media

- [Interview Experiences](#)
- [Advanced Data Structures](#)
- [Dynamic Programming](#)
- [Greedy Algorithms](#)
- [Backtracking](#)
- [Pattern Searching](#)
- [Divide & Conquer](#)
- [Mathematical Algorithms](#)
- [Recursion](#)
- [Geometric Algorithms](#)

• Popular Posts

- [All permutations of a given string](#)
- [Memory Layout of C Programs](#)
- [Understanding “extern” keyword in C](#)
- [Median of two sorted arrays](#)
- [Tree traversal without recursion and without stack!](#)
- [Structure Member Alignment, Padding and Data Packing](#)
- [Intersection point of two Linked Lists](#)
- [Lowest Common Ancestor in a BST](#)
- [Check if a binary tree is BST or not](#)
- [Sorted Linked List to Balanced BST](#)

Follow @GeeksforGeeks



[Subscribe](#)

• Recent Comments

- [Bharath Kumar Reddy Janumpally](#)

findsubsets(int [] arr) { int numofsubsets =...

[Power Set](#) · [1 minute ago](#)

- [Siya](#)

We can do it in $O(n \log n)$ i guess by using...

[Myntra Interview Experience | Set 4 \(For Senior Software Engineer \)](#) · [2 minutes ago](#)

- [darkprotocol](#)

Thanks god u didnt selected for SE role. It...

[Amazon Interview | Set 124 \(On-Campus\)](#) · [22 minutes ago](#)

- [Humble_Learner](#)

When the range represented by the node (...)

[Segment Tree | Set 2 \(Range Minimum Query\)](#) · [34 minutes ago](#)

- [Humble_Learner](#)

Learned Something New! :)

[Segment Tree | Set 1 \(Sum of given range\)](#) · [43 minutes ago](#)

- [tiger](#)

i think efficient solution for the problem in...

[Myntra Interview Experience | Set 4 \(For Senior Software Engineer \)](#) · [44 minutes ago](#)

-

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks team