

Given only a pointer to a node to be deleted in a singly linked list, how do you delete it?

A **simple solution** is to traverse the linked list until you find the node you want to delete. But this solution requires pointer to the head node which contradicts the problem statement.

Fast solution is to copy the data from the next node to the node to be deleted and delete the next node. Something like following.

```
struct node *temp = node_ptr->next;
node_ptr->data = temp->data;
node_ptr->next = temp->next;
free(temp);
```

Program:

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

/* Link list node */
struct node
{
    int data;
    struct node* next;
};

/* Given a reference (pointer to pointer) to the head
of a list and an int, push a new node on the front
of the list. */
void push(struct node** head_ref, int new_data)
{
```

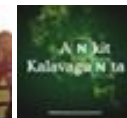
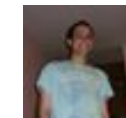
Google™ Custom Search



GeeksforGeeks



53,528 people like [GeeksforGeeks](#).



[Interview Experiences](#)

[Advanced Data Structures](#)

[Dynamic Programming](#)

[Greedy Algorithms](#)

[Backtracking](#)

[Pattern Searching](#)

[Divide & Conquer](#)

[Mathematical Algorithms](#)

[Recursion](#)

Free Online Database

 zoho.com/creator/online-database

Create database apps in minutes.
Just Drag-&-drop. Try NOW!



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](#)

```

/* 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;
}

void printList(struct node *head)
{
    struct node *temp = head;
    while(temp != NULL)
    {
        printf("%d ", temp->data);
        temp = temp->next;
    }
}

void deleteNode(struct node *node_ptr)
{
    struct node *temp = node_ptr->next;
    node_ptr->data = temp->data;
    node_ptr->next = temp->next;
    free(temp);
}

/* Driver program to test above function*/
int main()
{
    /* Start with the empty list */
    struct node* head = NULL;

    /* Use push() to construct below list
    1->12->1->4->1 */
    push(&head, 1);
    push(&head, 4);
    push(&head, 1);
    push(&head, 12);
    push(&head, 1);

    printf("\n Before deleting \n");

```

```

printList(head);

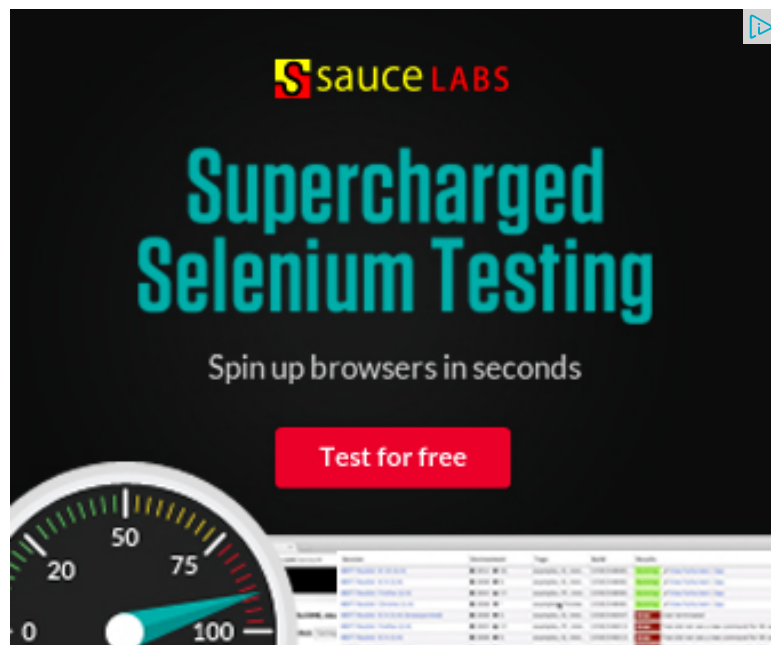
/* I m deleting the head itself.
   You can check for more cases */
deleteNode(head);

printf("\n After deleting \n");
printList(head);
getchar();
}

```

This solution doesn't work if the node to be deleted is the last node of the list. To make this solution work we can mark the end node as a dummy node. But the programs/functions that are using this function should also be modified.

Try this problem for doubly linked list.



Related Topics:

- 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
- Delete N nodes after M nodes of a linked list
- Design a stack with operations on middle element
- Swap Kth node from beginning with Kth node from end in a Linked List



1



Tweet

0



1

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

18 Comments

GeeksforGeeks

Sort by Newest ▾



Join the discussion...



chakradhar reddy · 6 hours ago

This will not work if its the last node right? Because we need to make the next deleting th enode.

^ | ▾ · Reply · Share ›



Himanshu Dagar · 3 months ago

can refer below link for this

<http://ideone.com/dc2uVv>

^ | ▾ · Reply · Share ›



Himanshu Dagar · 3 months ago

In Doubly linked list no need of copying
b/c it allows Bidirectinal iteration

705



Subscribe

Recent Comments

Abhi You live US or India?

[Google \(Mountain View\) interview](#) · 47 minutes ago

Aman Hi, Why arent we checking for conditions...

[Write a C program to Delete a Tree.](#) · 1 hour ago

kzs please provide solution for the problem...

[Backtracking | Set 2 \(Rat in a Maze\)](#) · 1 hour ago

Sanjay Agarwal bool

tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

[Find subarray with given sum](#) · 2 hours ago

AdChoices ▶

▶ [Linked List](#)

▶ [C++ Code](#)

▶ [Java Source Code](#)

cc: @sandeep, @prateek, @hanny, @NitinnPallindrome, @GeeksforGeeks, @payalgoel

^ | v • Reply • Share ›



sandeep • 6 months ago

This should not be called deleting node..

As this is just copying next node...

Deleting node means free the memory of the node at that address not the next

2 ^ | v • Reply • Share ›



prateek • a year ago

I guess there is no way to delete last node if only pointer to that node is given.
don't seem to work too.

@admin: correct if I'm wrong.

4 ^ | v • Reply • Share ›



hanny → prateek • 8 months ago

let suppose last node is ptr so next of ptr is null so we can put in ptr eq
null if ptr is null when we print the linked list for output next of previous is
null...last node is delete

1 ^ | v • Reply • Share ›



Nitin Pallindrome → hanny • 8 months ago

Dude we are copying the data of next node to the given node ...
which is null ...means not pointing to a place

^ | v • Reply • Share ›



GeeksforGeeks → prateek • a year ago

Agree with you.

^ | v • Reply • Share ›



payalgoel → GeeksforGeeks • 10 months ago

AdChoices ▶

▶ [C Null Pointer](#)

▶ [Linked Data](#)

▶ [Node 2](#)

AdChoices ▶

▶ [Programming C++](#)

▶ [Void Pointer](#)

▶ [Pointers Pointer](#)

admin I guess the code given by mayank / agarwal is working
Correct me if i'm wrong.

^ | v • Reply • Share ›



GeeksforGeeks Mod → payalgoel • 9 months ago

The solution suggested by Mayank looks good if double
the question says pointer to a node, not a double pointer

^ | v • Reply • Share ›



samrat • a year ago

Hi,

Here in this approach the given node is deleted by copying the data of the next
be a problem if the node has a lot of variables right?

like if the node is a student for example

node student

```
{  
String name;  
int id;  
int marks;  
String grade;  
.....  
.....  
..  
}
```

So, isnt there a better solution??

^ | v • Reply • Share ›



mayank agarwal • 2 years ago

this code works for last node deletion also...enjoy

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

void deleteNode(struct node **node)
{
    struct node *prev, *current=*node;

    if((*node)->next!=NULL)
    {
        while(current->next != NULL)
        { prev=current;
```

[see more](#)

^ | v • Reply • Share ›



deepuanand → mayank agarwal • 13 days ago

deleteNode function can be replaced with following. No need to go till the

```
void deleteNode(struct node **node)
{
    struct node *tmp,*current;
    if(node == NULL)
    {
        printf("give a valid pointer\n");
        return;
    }
```

```
current->next,  
if(current->next!=NULL)  
{  
tmp = current->next;  
current->data = tmp->data;  
current->next = tmp->next;  
node=&tmp;  
}  
free(*node);  
*node=NULL;  
}
```

^ | v • Reply • Share ›



Ajinkya • 2 years ago

Mark the node as a dummy node ?
What does that mean.?

^ | v • Reply • Share ›



GeeksforGeeks • 4 years ago

@nesamani1822:

The code looks fine now.

We cannot delete last node if only pointer to a node to be deleted is given. We
to change next of previous node to NULL,

If the last node is a dummy node and it is guaranteed that last node is never pa
approaches given will work.

^ | v • Reply • Share ›



nesamani1822 • 4 years ago

Please find the code with correction. I think this will work fine.


```

void deleteNode(struct node *node)
{
    struct node *prev;
    while(node->next != NULL)
    {
        prev = node;
        node->data = node->next->data;
        node = node->next;
    }
    free(node);
    prev->next = NULL;
}

```

This will work if the deleting node is not the last node. Can anyone tell me how

1 ^ | v • Reply • Share ›



GeeksforGeeks • 4 years ago

@nesamani1822: Thanks for suggesting a new approach, but there seems to

If we set node = NULL at the end of the loop, it will just set the local variable of free() is not called inside deleteNode()

You can try below code, instead of deleting 20 it modifies the list to 10->30->30

```

#include<stdio.h>

/* Link list node */
struct node
{
    int data;
    struct node* next;
}

```

```
};
```

```
void deleteNode(struct node *node)
{
```

[see more](#)

^ | v • Reply • Share ›



nesamani1822 • 4 years ago

This method is simple to delete using the pointer of the node to delete.

Steps:

- 1) move the next node's data to current node's data and do it till the last node
- 2) delete the last node

```
void deleteNode(struct node *node_ptr)
{
    while(node->next !=null)
    {
        node->data = node ->next->data;
        node = node->next;
    }
    node = null;
}
```

Example:

1->2->3->4->5 and node to delete is 3

1->2->4->4->5

1->2->4->5->5

now delete 5

 [Subscribe](#)

 [Add Disqus to your site](#)

@geeksforgeeks, **Some rights reserved**

[Contact Us!](#)

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team