Write a function to get Nth node in a Linked List

Write a GetNth() function that takes a linked list and an integer index and returns the data value stored in the node at that index position.

Algorithm:

```
1. Initialize count = 0
2. Loop through the link list
     a. if count is equal to the passed index then return current
         node
     b. Increment count
     c. change current to point to next of the current.
```

Implementation:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.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)
{
    /* 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;
}

/* Takes head pointer of the linked list and index
    as arguments and return data at index*/
int GetNth(struct node* head, int index)
{
    struct node* current = head;
    int count = 0; /* the index of the node we're currently
                    looking at */
    while (current != NULL)
    {
       if (count == index)
          return(current->data);
       count++;
       current = current->next;
    }

    /* if we get to this line, the caller was asking
       for a non-existent element so we assert fail */
    assert(0);
}

/* Drier 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);
```


Popular Posts

```
    /* Check the count function */
    printf("Element at index 3 is %d", GetNth(head, 3));
    getchar();
}
```

Time Complexity: O(n)

Related Tpoics:

- 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

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

27 Comments GeeksforGeeks

705 Subscribe

Sort by Newest

Join the discussion…

Himanshu Dagar • 3 months ago
http://ideone.com/U646FI

can refer to this code

Reply • Share ›

Himanshu Dagar • 3 months ago
(y)

Reply • Share ›

Himanshu Dagar • 3 months ago
liked
great job!!!

Reply • Share ›

kuldeep tripathi • 4 months ago
#include<iostream>

#include <algorithm>

#include <cstdio>

Recent Comments

Abhi You live US or India?
Google (Mountain View) interview · 46 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

using namespace std;

struct node

{

int info;

struct node *link;

};

class LinkedList

{

see more

1 ^ | v • Reply • Share ›

Rishi • 7 months ago

It is actually a wrong implementation of linked list as it should insert nodes in e

9 ^ | v • Reply • Share ›

Kartik ➚ Rishi • 4 months ago

A node can either be inserted at the beginning or end. The main questi

3 ^ | v • Reply • Share ›

sandeep • 10 months ago

#include
#include

typedef struct binarynode
{
int data;

```
};

binarynode * create(int value)
{
binarynode * root = (binarynode *)malloc(sizeof(binarynode));
root->data=value;
root->next=NULL;
return root;
}

void printlist(binarynode * root
)
```

see more

1 ^ | v • Reply • Share ›

Piyush Gandhi • 11 months ago

@GeeksForGeeks ....explain me one thing please:
The code you wrote here creates a stack here and not a linked list so wont it b
Pushing 1,4,1,12,1 into linked list and finding 2nd node data
must give ans as 4
but according to your method gives answer as 12..?

8 ^ | v • Reply • Share ›

morth • 11 months ago

My version, continue to iterate over linked list from last position, if possible. Re
head is NULL

```
 node_ref getn(const node_ref head, const int n)
{
  static int p = 0;
  static node_ref node = NULL;
```

```
  if (p > n)
  {
    p = 0;
    node = head;
  }

  while(n != p && node != NULL)
  {
    node = node->next;
    p++;
  }

  return node;
}
```

^ | v • Reply • Share ›

Ankit Malhotra • a year ago

Keep it simple :) The below method will do just what is needed and return NUL fuzzing around n is unsigned so that negative values cant be passed. For n =

```
node * nthnode (node * ptr, unsigned n) {
 if (!n) return ptr;
 while (--n && ptr) ptr = ptr->next;
 return ptr;
}
```

4 ^ | v • Reply • Share ›

Animesh Pratap Singh Sikarwar • a year ago

```
void Nth(struct node* head, int n)
{
```

```
    struct node* mover=head;
    while(mover->next!=NULL&&n!=1)
    {
        mover=mover->next;
        n--;
    }
    if(n==1)
    {
        printf("%d",mover->info);
    }
    else
    printf("limit exceeded");

}
```


Nikin Kumar Jain • a year ago

Was asked to give a perfect answer at Amazon. This function will handle all th

```
node* getNth(node *sr, int n)
{
        if(sr == NULL)
                return NULL;
        while(--n)
        {
                if(sr->next)
                        sr = sr->next;
                else
                        return NULL;
```

```
        return sr;
}
```

3 ▲ | ▼ • Reply • Share ›

Ramesh.Mxian ➦ Nikin Kumar Jain • a year ago

Will it work if n is given 0 or <0

1 ▲ | ▼ • Reply • Share ›

nikinjain ➦ Ramesh.Mxian • 4 months ago

I think this should be a good solution.

```
void getNth(node *head, int num)

{

if(!head || num < 0)

std::cout<<"Invalid Data";

if(num == 0)

std::cout<<head->data;

while(--num)

{

head = head->next;

if(!head)

std::cout<<"Invalid Number Value";
```

}

std::cout<<head->data;

}

Reply • Share ›

nikinjain ➔ Ramesh.Mxian • 4 months ago

Thanks for pointing out case

Reply • Share ›

Anand ➔ nikinjain • 17 days ago

what if argument num has value more than the length o

consider num = 2 and length(linklist) = 1

Reply • Share ›

Ankit Malhotra ➔ Nikin Kumar Jain • a year ago

How about

```
node* getNth(node *sr, int n)
{
    while(sr && --n) sr = sr->next;
    return sr;
}
```

Reply • Share ›

Somananda Ningom • a year ago

That&#039s the program I need....good.

Manoj Sharma • a year ago

```
#include<stdlib.h>
#include<stdio.h>
void create();
void getnthnode();
struct node
{

int info;.

struct node *next;.

}*start=NULL;
int main()

{

char ch;.
do.
{.
```

see more

• Reply • Share ›

VR • 2 years ago

```
struct node* getNthNode(struct node *head, int index)
{
       int count = 2;
```

```
        return head;
    else
    {
        curr = head ->next;
        i = index;
        if(index is even)
        {
             if(i==count)
                    return curr;
             if(curr->next->next!=null)
             {
```

see more

• Reply • Share ›

aimless • 3 years ago

```
/* Takes head pointer of the linked list and index
    as arguments and return data at index*/
int GetNth(struct node* head, int index)
{
    while (index - 1 > 0 && head)
    {
       head = head->next;
       index--;
    }

    if(index-1==0 && head)
       return head->data;

    /* if we get to this line, the caller was asking
```

```
    assert(0);
}
```

• Reply • Share ›

Manish_Dawar • 3 years ago

Thanks..

• Reply • Share ›

ram • 3 years ago

This code will give incorrect answer if I pass a circular linked list or a linked list only 1 node and self looped.

1 • Reply • Share ›

GeeksforGeeks ➚ ram • 3 years ago

@ram: Yes, the code won't work for circular linked list. It assumes that write different functions for singly and circular linked lists in general.

• Reply • Share ›

Murat M Ozturk • 4 years ago

It is possible to reduce the number of node access to O(logn) if we use binary

1 • Reply • Share ›

foobar ➚ Murat M Ozturk • 3 years ago

your other option is skiplist.

1 • Reply • Share ›

KattyBlackyard • 5 years ago

Hi, gr8 post thanks for posting. Information is useful!

• Reply • Share ›