



A computer science portal for geeks

Practice

IDE

Q&A

GeeksQuiz

## 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.

Example:

Input: 1->10->30->14, index = 2

Output: 30

The node at index 2 is 30

### 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:

#### C

```
// C program to find n'th node in linked list
#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);
}

/* 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);
}

```

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

## Java

```
// Java program to find n'th node in linked list
class LinkedList
{
    Node head; //the head of list

    class Node
    {
        int data;
        Node next;
        Node(int d)
        {
            data = d;
            next = null;
        }
    }

    /* Takes index as argument and return data at index*/
    public int GetNth(int index)
    {
        Node current = head;
        int count = 0; /* index of Node we are
                        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 */
        System.out.println("Access failed");
        return 0;
    }

    /* Given a reference to the head of a list and an int,
    inserts a new Node on the front of the list. */
    public void push(int new_data)
    {

```

```
/* 1. alloc the Node and put data*/
Node new_Node = new Node(new_data);

/* 2. Make next of new Node as head */
new_Node.next = head;

/* 3. Move the head to point to new Node */
head = new_Node;
}

/* Drier program to test above functions*/
public static void main(String[] args)
{
    /* Start with empty list */
    LinkedList llist = new LinkedList();

    /* Use push() to construct below list
       1->12->1->4->1 */
    llist.push(1);
    llist.push(4);
    llist.push(1);
    llist.push(12);
    llist.push(1);

    /* Check the count function */
    System.out.println("Element at index 3 is "+llist.GetNth(3));
}
}
```

Output:

```
Element at index 3 is 4
```

**Time Complexity:**  $O(n)$

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



128 Comments Category: [Linked Lists](#) Tags: [GetNth](#), [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)

1

Average Difficulty : **1/5.0**  
Based on **9** vote(s)

☐

Add to TODO List

☐

Mark as DONE

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!](#)