



C++ Program For Finding The Middle Element Of A Given Linked List

Last Updated : 17 Aug, 2023

Given a singly linked list, find the middle of the linked list.

1. For example, if the given linked list is 1->2->3->4->5 then the output should be 3.

If there are even nodes, then there would be two middle nodes, we need to print the second middle element.

2. For example, if given linked list is 1->2->3->4->5->6 then the output should be 4.

Method 1:

Traverse the whole linked list and count the no. of nodes.

Now traverse the list again till count/2 and return the node at count/2.

C++

```
// C++ program for the above approach

#include <iostream>
using namespace std;
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```

        Node* next;
    };

class NodeOperation {
public:
    // Function to add a new node
    void pushNode(class Node** head_ref, int data_val)
    {

        // Allocate node
        class Node* new_node = new Node();

        // Put in the data
        new_node->data = data_val;

        // Link the old list of the new node
        new_node->next = *head_ref;

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

    // A utility function to print a given linked list
    void printNode(class Node* head)
    {
        while (head != NULL) {
            cout << head->data << "->";
            head = head->next;
        }
        cout << "NULL" << endl;
    }

    /* Utility Function to find length of linked list */
    int getLen(class Node* head)
    {
        int len = 0;
        class Node* temp = head;
        while (temp) {
            len++;
            temp = temp->next;
        }
        return len;
    }
}

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

```

    if (head) {
        // find length
        int len = getLen(head);
        class Node* temp = head;

        // traverse till we reach half of length
        int midIdx = len / 2;
        while (midIdx--) {
            temp = temp->next;
        }
        // temp will be storing middle element
        cout << "The middle element is [" << temp->data
            << "]" << endl;
    }
}

// Driver Code
int main()
{
    class Node* head = NULL;
    class NodeOperation* temp = new NodeOperation();
    for (int i = 5; i > 0; i--) {
        temp->pushNode(&head, i);
        temp->printNode(head);
        temp->printMiddle(head);
    }
    return 0;
}

```

Output

```

5->NULL
The middle element is [5]
4->5->NULL
The middle element is [5]
3->4->5->NULL
The middle element is [4]
2->3->4->5->NULL

```

Time Complexity: O(n) where n is no of nodes in linked list

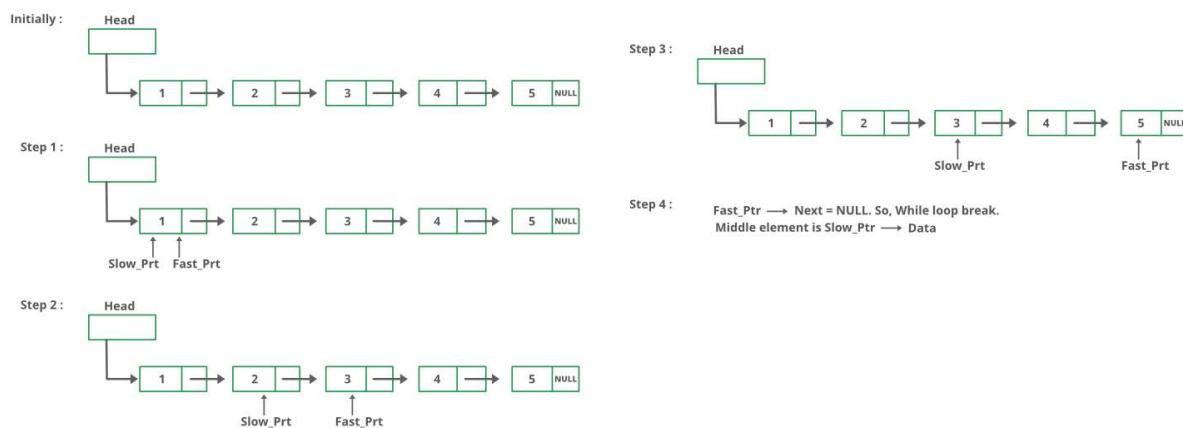
Auxiliary Space: O(1)

Method 2:

Traverse linked list using two pointers.

Move one pointer by one and the other pointers by two. When the fast pointer reaches the end slow pointer will reach the middle of the linked list.

Below image shows how printMiddle function works in the code :



C++

```
// C++ program for the above approach
```

```
#include <iostream>
using namespace std;

class Node {
public:
    int data;
    Node* next;
};

class NodeOperation {
public:
    // Function to add a new node
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

```

class Node* new_node = new Node();

// Put in the data
new_node->data = data_val;

// 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;
}

// A utility function to print a given linked list
void printNode(class Node* head)
{
    while (head != NULL) {
        cout << head->data << "->";
        head = head->next;
    }
    cout << "NULL" << endl;
}

void printMiddle(class Node* head)
{
    struct Node* slow_ptr = head;
    struct Node* fast_ptr = head;

    if (head != NULL) {
        while (fast_ptr != NULL
                && fast_ptr->next != NULL) {
            fast_ptr = fast_ptr->next->next;
            slow_ptr = slow_ptr->next;
        }
        cout << "The middle element is ["
            << slow_ptr->data << "]"
            << endl;
    }
}

// Driver Code
int main()
{
    class Node* head = NULL;
}

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

```

        temp->printMiddle(head);
    }
    return 0;
}

```

Output

```

5->NULL
The middle element is [5]
4->5->NULL
The middle element is [5]
3->4->5->NULL
The middle element is [4]
2->3->4->5->NULL
The middle element is [4]
1->2->3->4->5->NULL
The middle element is [3]

```

Time Complexity: O(n) where n is the number of nodes in the given linked list.

Auxiliary Space: O(1), no extra space is required, so it is a constant.

Method 3:

Initialize mid element as head and initialize a counter as 0. Traverse the list from head, while traversing increment the counter and change mid to mid->next whenever the counter is odd. So the mid will move only half of the total length of the list.

Thanks to Narendra Kangralkar for suggesting this method.

C++

```
#include <bits/stdc++.h>
using namespace std;
```

```

};

// Function to get the middle of
// the linked list
void printMiddle(struct node* head)
{
    int count = 0;
    struct node* mid = head;

    while (head != NULL) {

        // Update mid, when 'count'
        // is odd number
        if (count & 1)
            mid = mid->next;

        ++count;
        head = head->next;
    }

    // If empty list is provided
    if (mid != NULL)
        printf("The middle element is [%d]\n", mid->data);
}

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;
}

```

```

while (ptr != NULL) {
    printf("%d->", ptr->data);
    ptr = ptr->next;
}
printf("NULL\n");
}

// Driver code
int main()
{
    // Start with the empty list
    struct node* head = NULL;
    int i;

    for (i = 5; i > 0; i--) {
        push(&head, i);
        printList(head);
        printMiddle(head);
    }
    return 0;
}

// This code is contributed by ac121102

```

Output

```

5->NULL
The middle element is [5]
4->5->NULL
The middle element is [5]
3->4->5->NULL
The middle element is [4]
2->3->4->5->NULL
The middle element is [4]
1->2->3->4->5->NULL
The middle element is [3]

```

Please refer complete article on [Find the middle of a given linked list](#) for more details!

Join [GfG 160](#), a 160-day journey of coding challenges aimed at sharpening your skills. Each day, solve a handpicked problem, dive into detailed solutions through articles and videos, and enhance your preparation for any interview—all for free! Plus, win exciting GfG goodies along the way! - [Explore Now](#)



GeeksforGeeks



5

Next Article

C++ Program For Moving Last Element
To Front Of A Given Linked List

Similar Reads

[Javascript Program For Finding The Middle Element Of A Given Linked...](#)

Given a singly linked list, find the middle of the linked list. For example, if the given linked list is 1->2->3->4->5 then the output should be 3. If there are...

4 min read

[C++ Program For Finding The Length Of Longest Palindrome List In A...](#)

Given a linked list, find the length of the longest palindrome list that exists in that linked list. Examples: Input : List = 2->3->7->3->2->12->24 Output : ...

3 min read

[Javascript Program For Finding The Length Of Longest Palindrome Lis...](#)

Given a linked list, find the length of the longest palindrome list that exists in

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

C++ Program For Removing Middle Points From a Linked List Of Line...

Given a linked list of coordinates where adjacent points either form a vertical line or a horizontal line. Delete points from the linked list which are in the...

4 min read

Javascript Program For Removing Middle Points From a Linked List Of...

Given a linked list of coordinates where adjacent points either form a vertical line or a horizontal line. Delete points from the linked list which are in the...

4 min read

C++ Program For Making Middle Node Head In A Linked List

Given a singly linked list, find middle of the linked list and set middle node of the linked list at beginning of the linked list. Examples: Input: 1 2 3 4 5...

3 min read

C++ Program To Delete Middle Of Linked List

Given a singly linked list, delete the middle of the linked list. For example, if the given linked list is 1->2->3->4->5 then the linked list should be modifi...

4 min read

C++ Program For Inserting Node In The Middle Of The Linked List

Given a linked list containing n nodes. The problem is to insert a new node with data x at the middle of the list. If n is even, then insert the new node...

5 min read

Javascript Program For Making Middle Node Head In A Linked List

Given a singly linked list, find middle of the linked list and set middle node of the linked list at beginning of the linked list. Examples: Input: 1 2 3 4 5...

3 min read

4 min read

Article Tags :

C++

C++ Programs

DSA

Linked List

+17 More

Practice Tags :

Adobe

Amazon

Flipkart

GE

+14 More



Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddha
Nagar, Uttar Pradesh, 201305



Company

About Us

Legal

Careers

In Media

Contact Us

Advertise with us

GFG Corporate Solution

Placement Training Program

Explore

Job-A-Thon Hiring Challenge

Hack-A-Thon

GfG Weekly Contest

Offline Classes (Delhi/NCR)

DSA in JAVA/C++

Master System Design

Master CP

GeeksforGeeks Videos

Geeks Community

Languages

DSA

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

GoLang
SQL
R Language
Android Tutorial

DSA Roadmap
DSA Interview Questions
Competitive Programming

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
NodeJS
Bootstrap
Tailwind CSS

Python Tutorial

Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar

Commerce

Accountancy
Business Studies
Economics
Management
HR Management
Finance

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

MySQL
PostgreSQL
PL/SQL
MongoDB

Resume Templates
Aptitude Preparation
Puzzles
Company-Wise Preparation
Companies
Colleges

Competitive Exams

JEE Advanced
UGC NET
UPSC
SSC CGL
SBI PO
SBI Clerk
IBPS PO
IBPS Clerk

More Tutorials

Software Development
Software Testing
Product Management
Project Management
Linux
Excel
All Cheat Sheets
Recent Articles

Free Online Tools

Typing Test
Image Editor
Code Formatters
Code Converters
Currency Converter
Random Number Generator
Random Password Generator

Write & Earn

Write an Article
Improve an Article
Pick Topics to Write
Share your Experiences
Internships

DSA/Placements

DSA - Self Paced Course
DSA in JavaScript - Self Paced Course
DSA in Python - Self Paced
C Programming Course Online - Learn C with Data Structures
Complete Interview Preparation
Master Competitive Programming
Core CS Subject for Interview Preparation
Mastering System Design: LLD to HLD
Tech Interview 101 - From DSA to System Design [LIVE]
DSA to Development [HYBRID]
Placement Preparation Crash Course [LIVE]

Development/Testing

JavaScript Full Course
React JS Course
React Native Course
Django Web Development Course
Complete Bootstrap Course
Full Stack Development - [LIVE]
JAVA Backend Development - [LIVE]
Complete Software Testing Course [LIVE]
Android Mastery with Kotlin [LIVE]

Machine Learning/Data Science

Complete Machine Learning & Data Science Program - [LIVE]
Data Analytics Training using Excel, SQL, Python & PowerBI -

Programming Languages

C Programming with Data Structures
C++ Programming Course

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Clouds/Devops

DevOps Engineering
AWS Solutions Architect Certification
Salesforce Certified Administrator Course

GATE

GATE CS & IT Test Series - 2025
GATE DA Test Series 2025
GATE CS & IT Course - 2025
GATE DA Course 2025

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).