

# GeeksforGeeks

A computer science portal for geeks

## 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](#)

## Practice questions for Linked List and Recursion

Assume the structure of a Linked List node is as follows.

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

Explain the functionality of following C functions.

## 1. What does the following function do for a given Linked List?

```
void fun1(struct node* head)
{
    if(head == NULL)
        return;

    fun1(head->next);
    printf("%d ", head->data);
}
```

fun1() prints the given Linked List in reverse manner. For Linked List 1->2->3->4->5, fun1() prints 5->4->3->2->1.

## 2. What does the following function do for a given Linked List ?

```
void fun2(struct node* head)
{
    if(head== NULL)
        return;
    printf("%d ", head->data);

    if(head->next != NULL )
        fun2(head->next->next);
    printf("%d ", head->data);
}
```

fun2() prints alternate nodes of the given Linked List, first from head to end, and then from end to head. If Linked List has even number of nodes, then fun2() skips the last node. For Linked List 1->2->3->4->5, fun2() prints 1 3 5 3 1. For Linked List 1->2->3->4->5->6, fun2() prints 1 3 5 5 3 1.

Below is a complete running program to test above functions.

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

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

/* Prints a linked list in reverse manner */
void fun1(struct node* head)
{
    if(head == NULL)
        return;

    fun1(head->next);
```

```

    printf("%d ", head->data);
}

/* prints alternate nodes of a Linked List, first
   from head to end, and then from end to head. */
void fun2(struct node* start)
{
    if(start == NULL)
        return;
    printf("%d ", start->data);

    if(start->next != NULL )
        fun2(start->next->next);
    printf("%d ", start->data);
}

/* UTILITY FUNCTIONS TO TEST fun1() and fun2() */
/* 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;
}

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

    /* Using push() to construct below list
       1->2->3->4->5 */
    push(&head, 5);
    push(&head, 4);
    push(&head, 3);
    push(&head, 2);
    push(&head, 1);

    printf("\n Output of fun1() for list 1->2->3->4->5 \n");
    fun1(head);
}

```

```
printf("\n Output of fun2() for list 1->2->3->4->5 \n");
fun2(head);

getchar();
return 0;
}
```

Please write comments if you find any of the answers/explanations incorrect, or you want to share more information about the topics discussed above.

## 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](#)

Tags: [Recursion](#)

Like 7

Tweet 0

 0

Writing code in comment? Please use [ideone.com](http://ideone.com) and share the link here.

43 Comments

GeeksforGeeks

 Login▼

 Recommend

 Share

Sort by Newest▼



Join the discussion...



**Himanshu Sarraf** • 10 days ago

how recursion is taking place in func1()..plz explain

^ | v • Reply • Share ›



**Aditya** • 2 months ago

How does the return statement execute the recursion in reverse order?

^ | v • Reply • Share ›



**Aravind** • 4 months ago

i think it would be better if anyone can tell me how fun1 and fun2 works. I need the explanations for that

^ | v • Reply • Share ›



**Mayur** → Aravind • 2 months ago

For every recursion use a stack!!

^ | v • Reply • Share ›



**Techie1991** → Aravind • 3 months ago

Take a pen and a paper. Sit and trace it. The only way to learn recursion.

4 ^ | v • Reply • Share ›



**Shubham Aggarwal** • 5 months ago

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

I don't understand how above code link the list to the new node. How I see this is, new node's link part now contain a pointer to the head node.

^ | v • Reply • Share ›



**Anurag Singh** → Shubham Aggarwal • 5 months ago

I guess you are talking about "push" function.

Here the newly created node is made "head" of the existing list.

e.g. Lets say existing list is "4 ->3->2->1" and newly created node is "5".

(\*head\_ref) points to "4", the head of existing list and new\_node points to "5" and new\_node->next is NULL.

With above line, node->next points to (\*head\_ref), so the list becomes "5->4 ->3->2->1".

(\*head\_ref) still points to "4", so following line:

(\*head\_ref) = new\_node;

changes (\*head\_ref) to point to "5", i.e. the head of modified list.

1 ^ | v • Reply • Share ›



**Shubham Aggarwal** → Anurag Singh • 5 months ago

Got it. Thank you :)

^ | v • Reply • Share ›



**pavitraa** • 5 months ago

please explain the examples in detail,i cant under stand

^ | v • Reply • Share ›



**Anurag Singh** → pavitraa • 5 months ago

Please be specific, which part you don't understand. Put enough debug statements if you need, to visualize the flow.

^ | v • Reply • Share ›



**Ravi Saxena** • 6 months ago



Please made me clear that how 531 print..in linked

^ | v • Reply • Share ›



**Anurag Singh** → Ravi Saxena • 5 months ago

I guess you are talking about fun2.

Here we are traversing alternate nodes i.e. 1st node, 3rd node, 5th node, 7th node and so on.

Here we have two printf statement. One print happens for the current node and then we move on to the next alternate node. We keep moving further until we reach end of list.

Then it prints current node and then moves back to earlier calls.

In linked list, 1->2->3->4->5, traversal starts from node 1, then it goes to node 3 and then 5. So these nodes get printed as 1->3->5

Now we reach end of the list, so last printf executes and control moves back to previous calls i.e. from node 5 to node 3 and then to node 1. This prints 5->3->1.

So overall result is 1->3->5->5->3->1

2 ^ | v • Reply • Share ›



**neha** → Anurag Singh • 3 days ago

have u any other practical question like this

and plz can u tell me how 1-2-3-4-5 reverse 5-4-3-2-1.....

^ | v • Reply • Share ›



**Mohit Chaudhary** • 7 months ago

can anybody explain example number 1.

thanks

^ | v • Reply • Share ›



**Vaishal Shah** → Mohit Chaudhary • 6 months ago

linked list is like this. 1->2->3->4

now fun1 will continuously calling while last element doesn't reach and in our example while head=4 and head->next=null so it will return and go in back trace so now 3 will execute as well as 2 and 1 also will print

^ | v • Reply • Share ›



**gaurav mishra** → Vaishal Shah • 6 months ago

recursion uses stack

i think you got it

^ | v • Reply • Share ›



**rk** • 9 months ago

gud one.

1 ^ | v • Reply • Share ›



**pradeep kumar** • 9 months ago

What will be the time complexities of these two programs if implemented separately????

^ | v • Reply • Share ›



**ANA** → pradeep kumar • 9 months ago

$O(n)$

3 ^ | v • Reply • Share ›



**Himanshu Dagar** • a year ago

We can do operations in a recursion

1. While going forward

Or

2. While coming backward.

Good questions for brush up.....

^ | v • Reply • Share ›



**pratuksha** → Himanshu Dagar • 7 months ago

how can for coming backward...?

^ | v • Reply • Share ›



**Hinata Hyuga** → pratuksha • 2 months ago

after calling recursive function use print statement. the you accessing it backward, similarly if you access before calling recursive function the you are going forward

^ | v • Reply • Share ›



**neha** → pratuksha • 6 months ago

calling numbers are going into stack and when they encountered the return statement , it gets pop from the stack

^ | v • Reply • Share ›



**Hina** • a year ago

can anyone give me examples regarding linked list implementation in C++ like we have implementations of list in telephone directories..

can i hav such type of examples ????

plzzz if anyone have it then do post it

^ | v • Reply • Share ›



**vicky** → Hina • a year ago

Hi Hina, here is an example for link list implementation.

yes in c++ you can implement it too. suppose for an employee enrollment system, each employee has some set of data to be added in the system, but at the same time it is not necessary all employee have same amount of data so here we can use linked list To

necessary all employees have same amount of data so here we can use linked list. To add different amount of data for each employee. And even it is useful when you are not aware of how many employee should be added in the future. You can dynamically add as much employee as needed by the system

1 ^ | v • Reply • Share ›



**Hina** → vicky • a year ago

Thanks alot for ur help but here i got stuck that when we deal with different data sets then how we change our logic in functions am totally zero in this perspective....:(

^ | v • Reply • Share ›



**CP** • a year ago

nice Qn ...  
thanks

^ | v • Reply • Share ›



**mdev** • 2 years ago

Not able to comment....

^ | v • Reply • Share ›



**mdev** • 2 years ago

Nice Questions...

^ | v • Reply • Share ›



**Nabeel Mahmood** • 2 years ago

gud

^ | v • Reply • Share ›



**greek** • 2 years ago

okay

^ | v • Reply • Share ›



**PRAVEEN** • 3 years ago

want more questions...

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

3 ^ | v • Reply • Share ›



**hARRY** → PRAVEEN • 2 years ago

Thanks

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





^ | v • Reply • Share ›



**ajit** → hARRY • 2 years ago

Either way, time complexity is gonna be  $O(n)$ .

^ | v • Reply • Share ›



**hARRY** → PRAVEEN • 2 years ago

Which is better Recursion or iterative for reversing a linked list

1 ^ | v • Reply • Share ›



**vicky** → hARRY • a year ago

Recursion is overhead for the system as internally the system will convert it to iterative expression

2 ^ | v • Reply • Share ›



**anamika** • 3 years ago

func2() will throw segmentation fault in the case where number of entries in the list are even as head->next->next will ask for address of a node that does not exist.

The code should also check and return for head->next = NULL;

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

^ | v • Reply • Share ›



**Hinata Hyuga** → anamika • 2 months ago

nope, because we did check for (head->next->next != NULL)

^ | v • Reply • Share ›



**hARRY** → anamika • 2 years ago

Your are wrong!!

2 ^ | v • Reply • Share ›



**Hermoine** → hARRY • 2 years ago

Sorry You are wrong.....

^ | v • Reply • Share ›



**kartik** → anamika • 3 years ago

@anamika: Please take a closer look at the function. It calls fun2(start->next->next) only when start->next is not NULL.

```
if(start->next != NULL )
    fun2(start->next->next);
```

^ | v • Reply • Share ›



**Sunil** • 4 years ago

Nice questions to test our recursive skills.

^ | v • Reply • Share ›



**Gyan** • 4 years ago

Nice questions to brush up...Thanks.

^ | v • Reply • Share ›



Subscribe



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...

[Mynta 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