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?

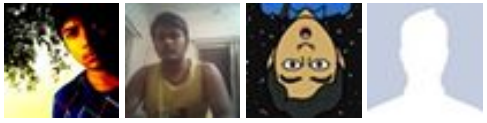
```
void fun1(struct node* head)
{
  if(head == NULL)
    return;

  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 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);
}
```

Geometric Algorithms

Popular Posts

```
/* 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.

share more information about the topics discussed above.

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.

20 Comments GeeksforGeeks

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 49 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

Sort by Newest

Join the discussion…

Himanshu Dagar • 17 days ago

We can do operations in a recursion
1. While going forward
Or
2. While coming backward.

Good questions for brush up.....

Reply • Share ›

Hina • 5 months ago

can anyone give me examples regarding linked list implementation in C++ like telephone directories..
can i hav such type of examples ????
plzzz if anyone have it then do post it

Reply • Share ›

vicky ➚ Hina • 4 months ago

Hi Hina, here is an example for link list implementation.
yes in c++ you can implement it too. suppose for an employee enrollm set of data to be added in the system, but at the same time it is not nec amount of data so here we can use linked list. To add different amount is useful when you are not aware of how many employee should be ad add as much employee as needed by the system

1 • Reply • Share ›

Hina ➚ vicky • 4 months ago

change our logic in functions am totally zero in this perspective

Reply • Share ›

CP • 6 months ago

nice Qn .. .
thanks

Reply • Share ›

mdev • 9 months ago

Not able to comment....

Reply • Share ›

mdev • 9 months ago

Nice Questions...

Reply • Share ›

Nabeel Mahmood • 9 months ago

gud

Reply • Share ›

greek • 10 months ago

okay

Reply • Share ›

PRAVEEN • 2 years ago

want more questions...

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

Reply • Share ›

hARRy ➔ PRAVEEN • a year ago

Thanks

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

▲ | ▼ • Reply • Share ›

ajit ➔ hARRy • a year ago

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

▲ | ▼ • Reply • Share ›

hARRY ➔ PRAVEEN • a year ago

Which is better Recurssion or iterative for reversing a linked list

▲ | ▼ • Reply • Share ›

vicky ➔ hARRY • 4 months ago

Recursion is overhead for the system as internally the system

▲ | ▼ • Reply • Share ›

anamika • 2 years ago

func2() will throw segmentation fault in the case where number of entries in th
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 c
```

▲ | ▼ • Reply • Share ›

hARRY ➔ anamika • a year ago

Your are wrong!!

▲ | ▼ • Reply • Share ›

Sorry You are wrong.....

• Reply • Share ›

kartik anamika • 2 years ago

@anamika: Please take a closer look at the function. It calls fun2(start-not NULL.

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

• Reply • Share ›

Sunil • 3 years ago

Nice questions to test our recursive skills.

• Reply • Share ›

Gyan • 3 years ago

Nice questions to brush up...Thanks.

• Reply • Share ›

Subscribe Add Disqus to your site