Write a recursive function to print reverse of a Linked List

Note that the question is only about printing the reverse. To reverse the list itself see this

Difficulty Level: Rookie

Algorithm

```
printReverse(head)
  1. call print reverse for hed->next
  2. print head->data
```

Implementation:

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

/* Link list node */
struct node
{
    int data;
    struct node* next;
};

/* Function to reverse the linked list */
void printReverse(struct node* head)
{
  // Base case
  if(head == NULL)
    return;

  // print the list after head node
  printReverse(head->next);

  // After everything else is printed, print head
  printf("%d  ", head->data);
}

/*UTILITY FUNCTIONS*/
/* Push a node to linked list. Note that this function
  changes the head */
void push(struct node** head_ref, char 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 pochar to the new node */
    (*head_ref)    = new_node;
}

/* Drier program to test above function*/
int main()
{

  struct node* head = NULL;

  push(&head, 1);
  push(&head, 2);
  push(&head, 3);
  push(&head, 4);
```

```
  printReverse(head);
  getchar();
}
```

Time Complexity: O(n)

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

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

munjal • a month ago

```
void printListRecursively(struct node *p)
{
if(p==NULL)
{
return;
}
else
{
printListRecursively(p->link);

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


nitin mendirata • 2 months ago

Can this question be solved in O(1) space complexity?

• Reply • Share ›

karunakar • 5 months ago

here is effient solution for linked list reverse using recursion

```
struct node *reverse(struct node *head)
{
struct node *temp;
if(head==NULL)
return head;
temp=reverse(head->next);
head->next->next=head;
head->next=NULL;
return temp;
}
```

• Reply • Share ›

Manika ➦ karunakar • 3 months ago

No need to reverse list . you just have to print list

2 • Reply • Share ›

<HoldOnLife!#> ➦ karunakar • 3 months ago

your solution is for reversing the linked list and here it print the reverse order only

2 • Reply • Share ›

Swati • 9 months ago

Please someone help me with the following code , where I am trying to print the reverse linked list.
I am trying to travel till last element , print it then remove it , and reduce the Linked List by 1 size.
But it is giving segmentation fault

```
void PrintReverse_t(node *head)

{

if (head != NULL)
{
node *p;

node *q;

p = head;
```

while (p->next != NULL)

---

see more

• Reply • Share ›

DS+Algo=Placement ➚ Swati • 8 months ago

See, when only one node i.e. head will left,

"while" loop will not execute(as p->next is NULL).
so after loop, the statement "q->next = NULL" will try to get executed but q is not pointing any node. That's why it is showing error.

Solution: Try to make another case where only one node is left.

Hope you understood. Or feel free to ask further.

1 • Reply • Share ›

Swati ➚ DS+Algo=Placement • 8 months ago

thanks it is working now .

• Reply • Share ›

DS+Algo=Placement ➚ Swati • 8 months ago

Good...Always feel free to ask me.

• Reply • Share ›

sdj • 10 months ago

logic to reverse print a circular linked list:

```
void circRev(struct node *head1)
{
if(head1->next==head)
{
cout << head1->data << "\n";
return;
}
circRev(head1->next);
cout << head1->data << "\n";
}
```

• Reply • Share ›

Codeguru • a year ago

complete c code to reverse a linked list

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

```
#include<malloc.h>

struct node
{
int data;
struct node * link;
};

void insert_beg(struct node ** head,int num)
{
struct node * temp;
temp=(struct node *)malloc(sizeof(struct node));
temp->data=num;
temp->link=*head;
*head=temp;
return;
```

see more

2 • Reply • Share ›

prathviraj • 2 years ago

Could you give a solution without using recursive function?

• Reply • Share ›

DS+Algo=Placement prathviraj • 8 months ago

and here is the case where iteration is not the alternative to recursion.

• Reply • Share ›

Abhay prathviraj • a year ago

/* ya linklist can be reversed using stack or simply array*/
first count the number of elements in the list, and make the maximum size of the stack equal to the sizeof the list. After theat push element on to the stack, after pushing all the element start popping and read the element. doing this you can pring the linklist in reverse order.
the simple array code is given below

```
i=0;

while(ptr->link!=NULL)
{
arr[i]=ptr->info;
ptr=ptr->link;
i++;

}
for(j=i-1;j>=0;j--)
```

printf("%d",arr[j]);

4 ▲ | ▼ • Reply • Share ›

debashis_deb ➚ Abhay • a month ago

the same thing happens with the recursive version ! it uses stack too !

▲ | ▼ • Reply • Share ›

Anon ➚ prathviraj • a year ago

will be sort of cheating but you can use a stack ;)

▲ | ▼ • Reply • Share ›

Scholastica Peter • 2 years ago

so good when using recursive.

▲ | ▼ • Reply • Share ›

Ashish Singh • 2 years ago

solve above prblm by using recursive and non recursive function.

▲ | ▼ • Reply • Share ›

code4fun • 3 years ago

Here it is in C

```
 typedef struct node
{
        int data;
        struct node* next;
}NODE,*PNODE;

PNODE reverseLinkList(PNODE head)
{
        PNODE temp ;
        if (!head || !head->next)
                return head;

        temp = reverseLinkList(head->next);
        head->next->next = head;
        head->next = NULL;
        return temp;
}
```

▲ | ▼ • Reply • Share ›

BlackMath • 3 years ago

A java program to actually reverse a linked list by recursion, not only printing it.

```
 /* Paste your code here (You may delete these lines if not writing code) */
class LNode
{
   int value;
   LNode next;

   LNode (int val)
   {
      value = val;
   }
}

public class ReverseLinkedListByRecursion
{
  public static void printList (LNode head)
  {
```

see more

2 • Reply • Share ›

- 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

- Goku

  They are considering 0 based indexing instead...

  Write a function to get Nth node in a Linked List · 38 minutes ago

- lebron

  since the array size is 5, it takes constant...

  K’th Smallest/Largest Element in Unsorted Array | Set 3 (Worst Case Linear Time) · 4 hours ago

- lebron

  merge sort

  K’th Smallest/Largest Element in Unsorted Array | Set 3 (Worst Case Linear Time) · 4 hours ago

- Shubham Sharma

  You saved my time :)

  Searching for Patterns | Set 2 (KMP Algorithm) · 5 hours ago

- Prakhar

  Why so many LOCs, if I'm not wrong (please...

  Largest Sum Contiguous Subarray · 5 hours ago

- Aayush Gupta

  For R4 Q3, Another solution would be to use a...

  Amazon Interview Experience | Set 168 · 6 hours ago