

GeeksQuiz

Computer science mock tests for geeks

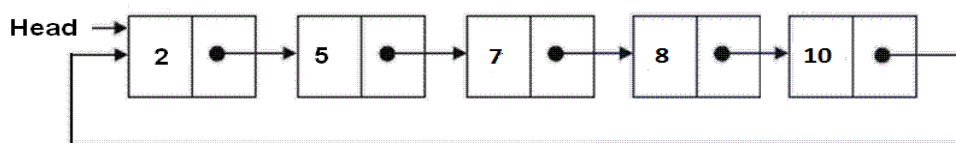
- [Home](#)
- [Latest Questions](#)
- [Articles](#)
- [C/C++ Programs](#)
- [Contribute](#)
- [Books](#)

[Subscribe](#)

Circular Linked List | Set 2 (Traversal)

April 10, 2014

We have discussed [Circular Linked List Introduction and Applications](#), in the previous post on Circular Linked List. In this post, traversal operation is discussed.



In a conventional linked list, we traverse the list from the head node and stop the traversal when we reach NULL. In a circular linked list, we stop traversal when we reach the first node again. Following is C code for linked list traversal.

```
/* Function to traverse a given Circular linked list and print nodes */
void printList(struct node *first)
{
    struct node *temp = first;

    // If linked list is not empty
    if (first != NULL)
    {
        // Keep printing nodes till we reach the first node again
        do
        {
            printf("%d ", temp->data);
            temp = temp->next;
        }
        while (temp != first);
    }
}
```

```

    }
}

```

Complete C program to demonstrate traversal. Following is complete C program to demonstrate traversal of circular linked list.

```

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

/* structure for a node */
struct node
{
    int data;
    struct node *next;
};

/* Function to insert a node at the begining of a Circular
linked list */
void push(struct node **head_ref, int data)
{
    struct node *ptr1 = (struct node *)malloc(sizeof(struct node));
    struct node *temp = *head_ref;
    ptr1->data = data;
    ptr1->next = *head_ref;

    /* If linked list is not NULL then set the next of last node */
    if (*head_ref != NULL)
    {
        while (temp->next != *head_ref)
            temp = temp->next;
        temp->next = ptr1;
    }
    else
        ptr1->next = ptr1; /*For the first node */

    *head_ref = ptr1;
}

/* Function to print nodes in a given Circular linked list */
void printList(struct node *head)
{
    struct node *temp = head;
    if (head != NULL)
    {
        do
        {
            printf("%d ", temp->data);
            temp = temp->next;
        }
        while (temp != head);
    }
}

```

```
/* Driver program to test above functions */
int main()
{
    /* Initialize lists as empty */
    struct node *head = NULL;

    /* Created linked list will be 12->56->2->11 */
    push(&head, 12);
    push(&head, 56);
    push(&head, 2);
    push(&head, 11);

    printf("Contents of Circular Linked List\n ");
    printList(head);

    return 0;
}
```

Output:

```
Contents of Circular Linked List
11 2 56 12
```

You may like to see following posts on Circular Linked List

[Split a Circular Linked List into two halves](#)

[Sorted insert for circular linked list](#)

We will soon be discussing implementation of insert delete operations for circular linked lists.

Please write comments if you find any bug in above code/algorithm, or find other ways to solve the same problem

Related Questions:

- [Binary Heap](#)
- [Delete all occurrences of a given key in a linked list](#)
- [How to create mergable stack?](#)
- [Deque | Set 1 \(Introduction and Applications\)](#)
- [A data structure for n elements and O\(1\) operations](#)
- [Convert left-right representation of a binary tree to down-right](#)
- [Print level order traversal line by line](#)
- [C Program for Red Black Tree Insertion](#)

Like 11

Tweet 1

+1 0

7 Comments

GeeksQuiz

Login ▾

♥ Recommend

🔗 Share

Sort by Best ▾



Join the discussion...



Guest • 3 months ago

in circular link list there is nothing as first node and last...we can start from any node given pointer to it....and traverse it untill we reach that node again....so last node will depend on which node you choose as first node....so any node can be first node and hence so is last node....which is dependent on first node....thankss

^ | v • Reply • Share ›



Rozal Agarwal • 5 months ago

Could it be possible that while traversing circular linked list, we reach to any of the traversed nodes again. I mean, it's not necessary that last node of linked list should point to the first node of linked list. It can point to any node of the circular linked list.

^ | v • Reply • Share ›



Ashish Jaiswal → Rozal Agarwal • 3 months ago

in circular link list there is nothing as first node and last...we can start from any node given pointer to it....and traverse it untill we reach that node again....so last node will depend on which node you choose as first node....so any node can be first node and hence so is last node....which is dependent on first node....thanksss

^ | v • Reply • Share ›



Kartik Nagpal • 6 months ago

@GeeksforGeeks - push function to insert a node in CLL has running time of $O(n)$ in above post. Instead, by simply maintaining a pointer to the last inserted node instead of the the first inserted node, running time of push can be reduced to $O(1)$.

By maintaining pointer to last inserted node, in CLL you can always get to the first inserted node("head") in $O(1)$ time by following the "next" of last inserted node("tail")

Also, one more thing i noticed in above main() function:

```
/* Created linked list will be 12->56->2->11 */
```

should be instead

```
/* Created linked list will be 11->2->56->12 */
```

I tweaked the above code in post to improve the insertion running time to $O(1)$ and it prints the list in same order as of insertion.

<http://ideone.com/BQaKPB>

^ | v • Reply • Share ›



ajayv • 8 months ago

```
void printList(struct node *head)
```



```
void printList(struct node *head)
{
    if(head == NULL) return;
    struct node *temp = head;
    while(temp->next != head){
        printf("%d ",temp->data);
        temp = temp->next;
    }
    printf("%d ",temp->data);
}
```

^ | v • Reply • Share ›



Amit Basak • 10 months ago

Why these two variables are declared when they are never used in main().

```
int list_size, i;
```

@admin: Should be removed any unused variable, as this will lead to compilation warning

^ | v • Reply • Share ›



GeeksforGeeks Mod ➔ Amit Basak • 8 months ago

Thanks for pointing this out. We have removed the unused variables.

^ | v • Reply • Share ›



Subscribe



Add Disqus to your site



Privacy



• Categories

- [Articles](#) (131)
 - [Algorithms](#) (24)
 - [C](#) (17)
 - [C++](#) (17)
 - [Data Structures](#) (29)
 - [DBMS](#) (1)
 - [Interview Experiences](#) (6)
 - [Java](#) (2)
 - [Operating Systems](#) (1)
 - [Puzzle](#) (18)
 - [Searching and Sorting](#) (10)
- [Programs](#) (35)
- [Quizzes](#) (2,107)
 - [Aptitude](#) (2)
 - [Computer Science Quizzes](#) (2,104)
 - [Algorithms](#) (147)
 - [C](#) (208)
 - [C++](#) (129)
 - [Computer Organization and Architecture](#) (1)
 - [Data Structures](#) (132)
 - [DBMS](#) (2)
 - [GATE](#) (1,406)
 - [Java](#) (51)
 - [Operating Systems](#) (28)
 - [Web technologies](#) (1)

• Recent Discussions

- [Ajoy](#)

In the dequeue function after dequeuing the...

[Queue | Set 1\(Introduction and Array Implementation\)](#) · [5 hours ago](#)

- [kaps](#)

can we solve the above using 1 pointer only...

[Linked List | Set 3 \(Deleting a node\)](#) · [6 hours ago](#)

- [kaps](#)

refer to previous post

[Linked List | Set 3 \(Deleting a node\)](#) · [6 hours ago](#)

- [Piyush](#)

Well,i am slightly confused at the part where...

[A Programmer's approach of looking at Array vs. Linked List](#) · [7 hours ago](#)

- Sandeep

Hey , can you please explain me about the array...

[C | Pointer Basics | Question 4](#) · [10 hours ago](#)

- [thevagabond85](#)

and for

Input: arr[] = {1, 1, 1, 1, 1, 1, 1}

[Count 1's in a sorted binary array](#) · [15 hours ago](#)

•

Valid [XHTML Strict 1.0](#)

Powered by [WordPress](#) & [MooTools](#) | MiniMoo 1.3.4