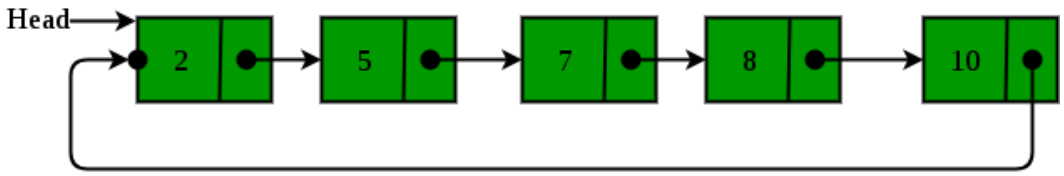
Circular Linked List Traversal in C++

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 are the codes for the linked list traversal.

```
/* Function to print nodes ina given Circular linked list */
void printList(Node* head)
{
    Node* temp = head;

    // If linked list is not empty
    if (head != NULL) {

        // Print nodes till we reach first node again
        do {
            cout << temp->data << " ";
            temp = temp->next;
        } while (temp != head);
    }
}
```

Complete program to demonstrate traversal. Following are complete programs to demonstrate traversal of circular linked list.

```
// C++ program to implement
// the above approach
#include <bits/stdc++.h>
using namespace std;

/* structure for a node */
class Node
{
    public:
    int data;
    Node *next;
};

/* Function to insert a node at the beginning
of a Circular linked list */
void push(Node **head_ref, int data)
{
    Node *ptr1 = new Node();
    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(Node *head)
{
    Node *temp = head;
    if (head != NULL)
    {
        do
        {
            cout << temp->data << " ";
            temp = temp->next;
        }
        while (temp != head);
    }
}

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

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

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

    return 0;
}
```

Output:

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