```
// C++ program to delete a given key from
// linked list.
#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)
{
	// Create a new node and make head as next
	// of it.
	Node* ptr1 = new Node();
	ptr1->data = data;
	ptr1->next = *head_ref;

	/* If linked list is not NULL then set the
	next of last node */
	if (*head_ref != NULL)
	{
		// Find the node before head and update
		// next of it.
		Node* temp = *head_ref;
		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);
        }

        cout << endl;
}

/* Function to delete a given node from the list */
void deleteNode(Node** head, int key)
{

        // If linked list is empty
        if (*head == NULL)
            return;
```

```
// If the list contains only a single node
if((*head)->data==key && (*head)->next==*head)
{
	free(*head);
	*head=NULL;
	return;
}

Node *last=*head,*d;

// If head is to be deleted
if((*head)->data==key)
{

	// Find the last node of the list
	while(last->next!=*head)
		last=last->next;

	// Point last node to the next of head i.e.
	// the second node of the list
	last->next=(*head)->next;
	free(*head);
	*head=last->next;
return;
}

// Either the node to be deleted is not found
// or the end of list is not reached
while(last->next!=*head&&last->next->data!=key)
{
	last=last->next;
```

```
    }

    // If node to be deleted was found
    if(last->next->data==key)
    {
        d=last->next;
        last->next=d->next;
        free(d);
    }
    else
        cout<<"no such keyfound";
    }

/* Driver code */
int main()
{
    /* Initialize lists as empty */
    Node* head = NULL;

    /* Created linked list will be 2->5->7->8->10 */
    push(&head, 2);
    push(&head, 5);
    push(&head, 7);
    push(&head, 8);
    push(&head, 10);

    cout << "List Before Deletion: ";
    printList(head);

    deleteNode(&head, 7);
```

```
    cout << "List After Deletion: ";
    printList(head);

    return 0;
}

// This is code is contributed by rathbhupendra
```