```
"""operation of the circular linked list"""
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class CircularSinglyLinkedList:
    def __init__(self):
        self.head = None

    def is_empty(self):
        return self.head is None

    def append(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = new_node
            new_node.next = self.head
        else:
            temp = self.head
            while temp.next != self.head:
                temp = temp.next
            temp.next = new_node
            new_node.next = self.head

    def insert_at_beginning(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = new_node
            new_node.next = self.head
        else:
            temp = self.head
            while temp.next != self.head:
                temp = temp.next
            new_node.next = self.head
            temp.next = new_node
            self.head = new_node

    def insert_at_end(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = new_node
            new_node.next = self.head
        else:
            temp = self.head
            while temp.next != self.head:
                temp = temp.next
            temp.next = new_node
            new_node.next = self.head

    def delete_node(self, key):
        if not self.head:
            print("List is empty")
            return
```

```
        temp = self.head
        prev = None

        # If the node to be deleted is the head
        if temp.data == key:
            while temp.next != self.head:
                temp = temp.next
            temp.next = self.head.next
            self.head = self.head.next
            return

        # Search for the node to be deleted
        while temp.next != self.head and temp.data != key:
            prev = temp
            temp = temp.next

        # If the node is not present
        if temp.data != key:
            print("Node not found")
            return

        # Delete the node
        prev.next = temp.next

    def search(self, key):
        if not self.head:
            print("List is empty")
            return False

        temp = self.head

        while temp.next != self.head:
            if temp.data == key:
                return True
            temp = temp.next

        # Check the last node
        if temp.data == key:
            return True

        return False

    def display(self):
        if self.is_empty():
            print("List is empty")
            return

        temp = self.head
        while True:
            print(temp.data, end=' -> ')
            temp = temp.next
            if temp == self.head:
                break
```

```
        print()

# Example usage:
if __name__ == "__main__":
    circular_list = CircularSinglyLinkedList()

    circular_list.append(1)
    circular_list.append(2)
    circular_list.append(3)
    circular_list.append(4)

    print("Original Circular Singly Linked List:")
    circular_list.display()

    circular_list.insert_at_beginning(0)
    print("\nAfter Inserting at Beginning:")
    circular_list.display()

    circular_list.insert_at_end(5)
    print("\nAfter Inserting at End:")
    circular_list.display()

    circular_list.delete_node(2)
    print("\nAfter Deleting Node with value 2:")
    circular_list.display()

    search_key = 3
    if circular_list.search(search_key):
        print(f"\nNode with value {search_key} found.")
    else:
        print(f"\nNode with value {search_key} not found.")

output
Original Circular Singly Linked List:
1 -> 2 -> 3 -> 4 ->

After Inserting at Beginning:
0 -> 1 -> 2 -> 3 -> 4 ->

After Inserting at End:
0 -> 1 -> 2 -> 3 -> 4 -> 5 ->

After Deleting Node with value 2:
0 -> 1 -> 3 -> 4 -> 5 ->

Node with value 3 found.
```