Circular Linked List in C

This document contains a C program demonstrating insertion and deletion operations in a Circular Linked List. Specifically, it shows how to insert and delete nodes at the beginning and at the end of the list.

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

struct Node {
    int data;
    struct Node* next;
};

struct Node* head = NULL;

// Function to insert a node at the beginning
void insertAtBeginning(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;

    if (head == NULL) {
        newNode->next = newNode;
        head = newNode;
        return;
    }

    struct Node* temp = head;
    while (temp->next != head) {
        temp = temp->next;
    }
    newNode->next = head;
    temp->next = newNode;
    head = newNode;
}

// Function to insert a node at the end
void insertAtEnd(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;

    if (head == NULL) {
        newNode->next = newNode;
        head = newNode;
        return;
    }
```

```
    struct Node* temp = head;
    while (temp->next != head) {
        temp = temp->next;
    }
    temp->next = newNode;
    newNode->next = head;
}

// Function to delete a node from the beginning
void deleteFromBeginning() {
    if (head == NULL) return;

    if (head->next == head) {
        free(head);
        head = NULL;
        return;
    }

    struct Node* temp = head;
    struct Node* last = head;

    while (last->next != head) {
        last = last->next;
    }
    head = head->next;
    last->next = head;
    free(temp);
}

// Function to delete a node from the end
void deleteFromEnd() {
    if (head == NULL) return;

    if (head->next == head) {
        free(head);
        head = NULL;
        return;
    }

    struct Node* temp = head;
    struct Node* prev = NULL;

    while (temp->next != head) {
        prev = temp;
        temp = temp->next;
    }
    prev->next = head;
    free(temp);
}

// Function to display the list
void display() {
    if (head == NULL) {
```

```
        printf("List is empty\n");
        return;
    }

    struct Node* temp = head;
    do {
        printf("%d -> ", temp->data);
        temp = temp->next;
    } while (temp != head);
    printf("(head)\n");
}

int main() {
    insertAtBeginning(10);
    insertAtEnd(20);
    insertAtBeginning(5);
    display();

    deleteFromBeginning();
    display();

    deleteFromEnd();
    display();

    return 0;
}
```