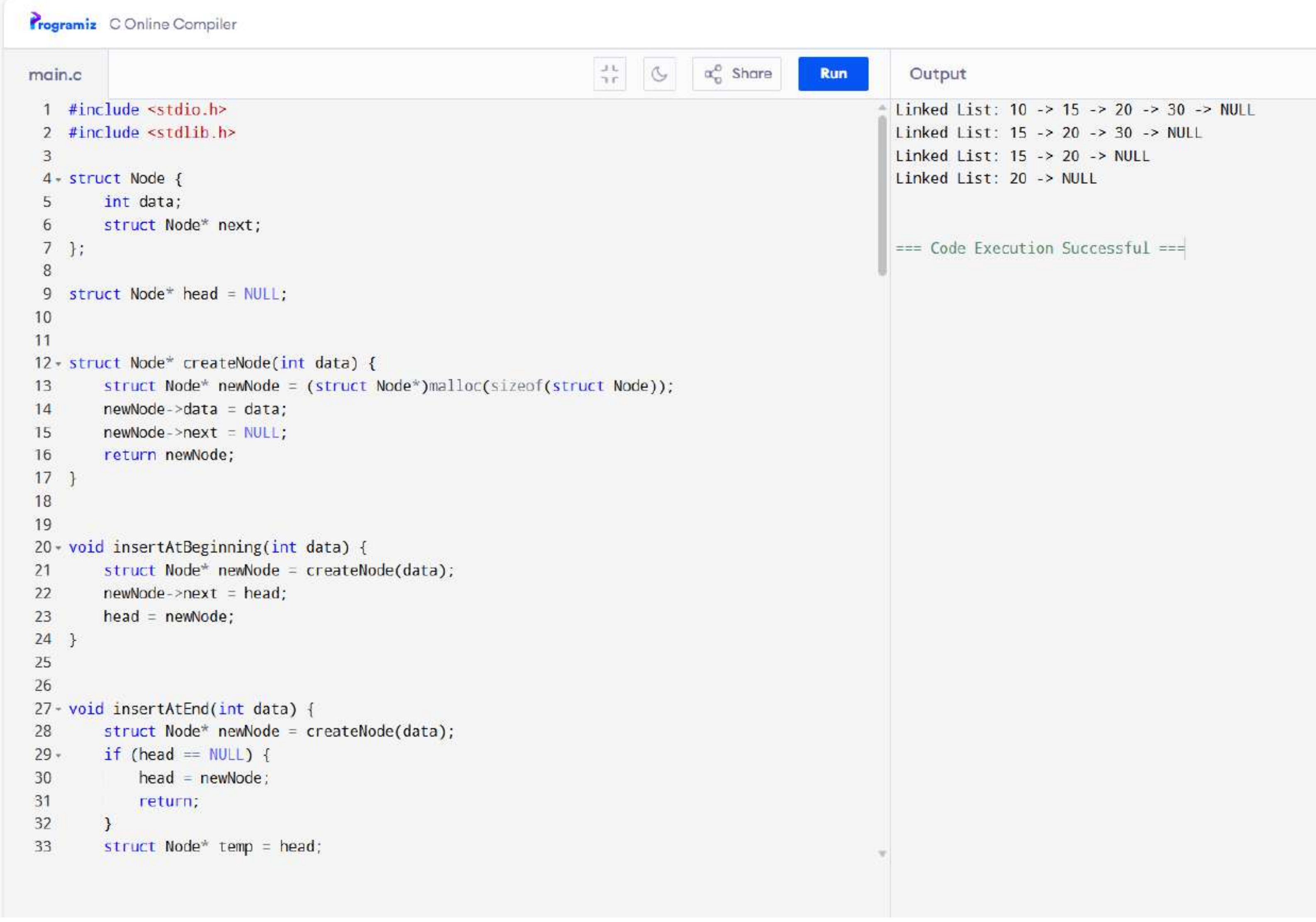
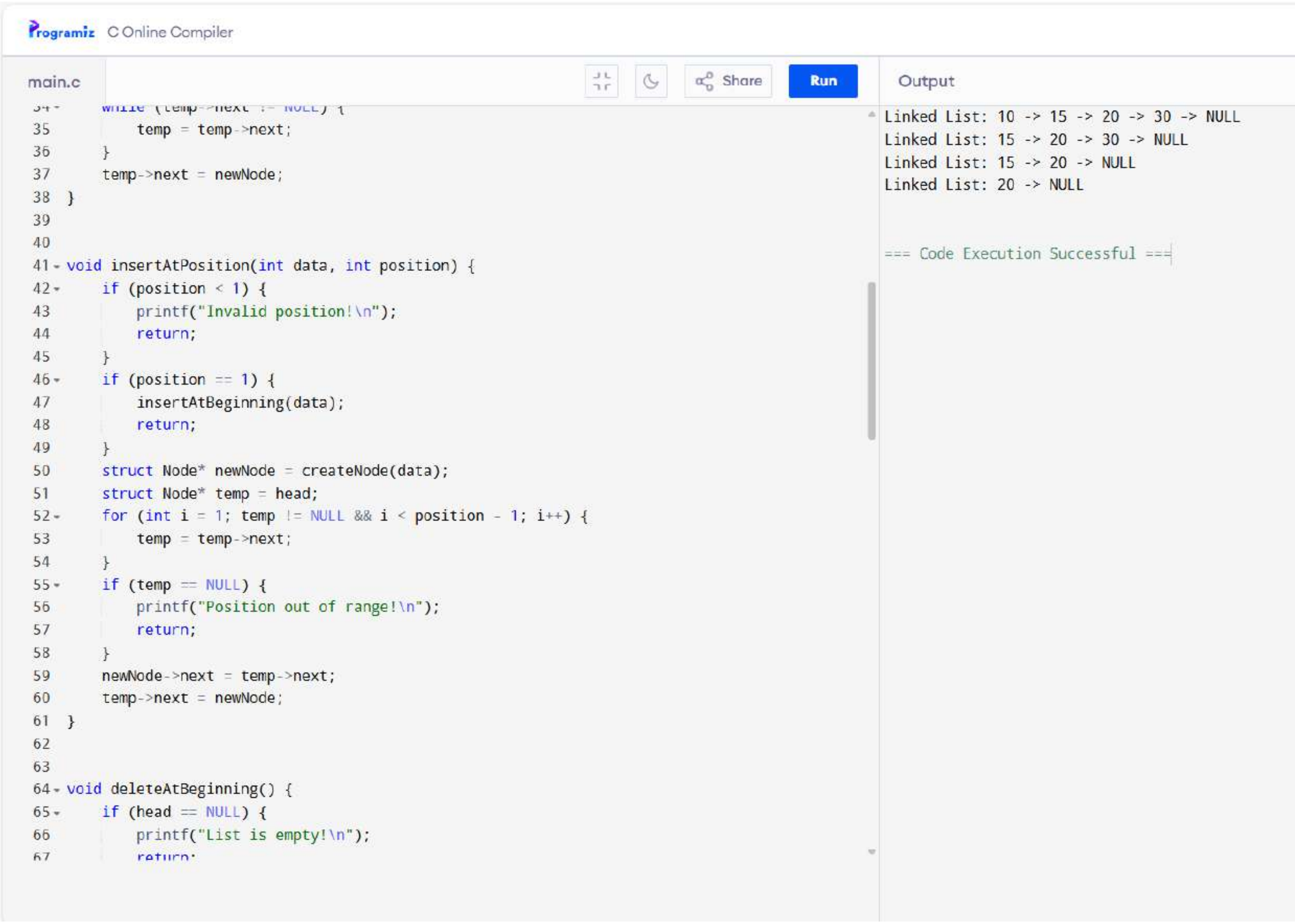


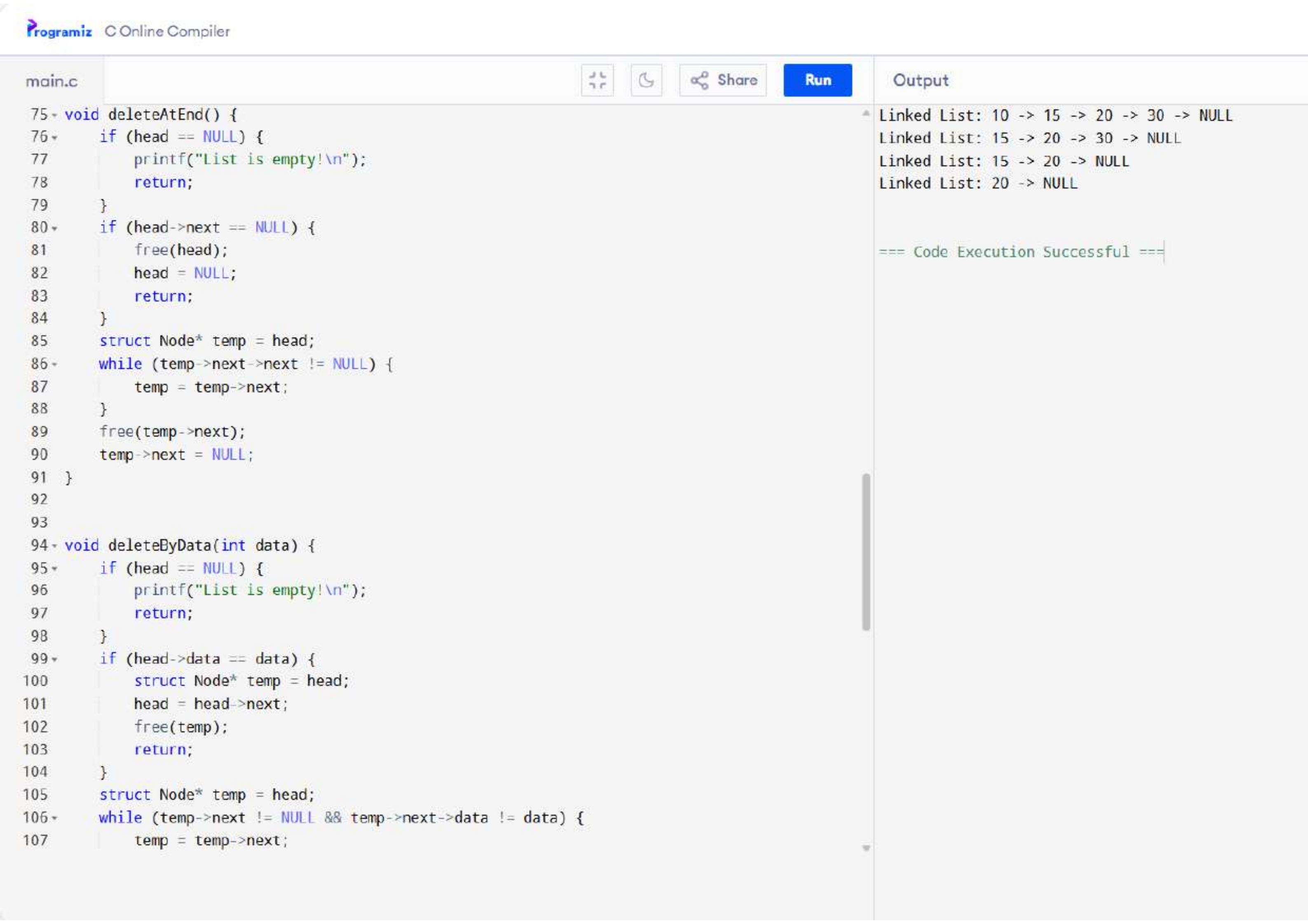
```c
#include <stdio.h>
#include <stdlib.h>

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

struct Node* head = NULL;


struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}


void insertAtBeginning(int data) {
    struct Node* newNode = createNode(data);
    newNode->next = head;
    head = newNode;
}


void insertAtEnd(int data) {
    struct Node* newNode = createNode(data);
    if (head == NULL) {
        head = newNode;
        return;
    }
    struct Node* temp = head;
```

```
Linked List: 10 -> 15 -> 20 -> 30 -> NULL
Linked List: 15 -> 20 -> 30 -> NULL
Linked List: 15 -> 20 -> NULL
Linked List: 20 -> NULL


=== Code Execution Successful ===
```



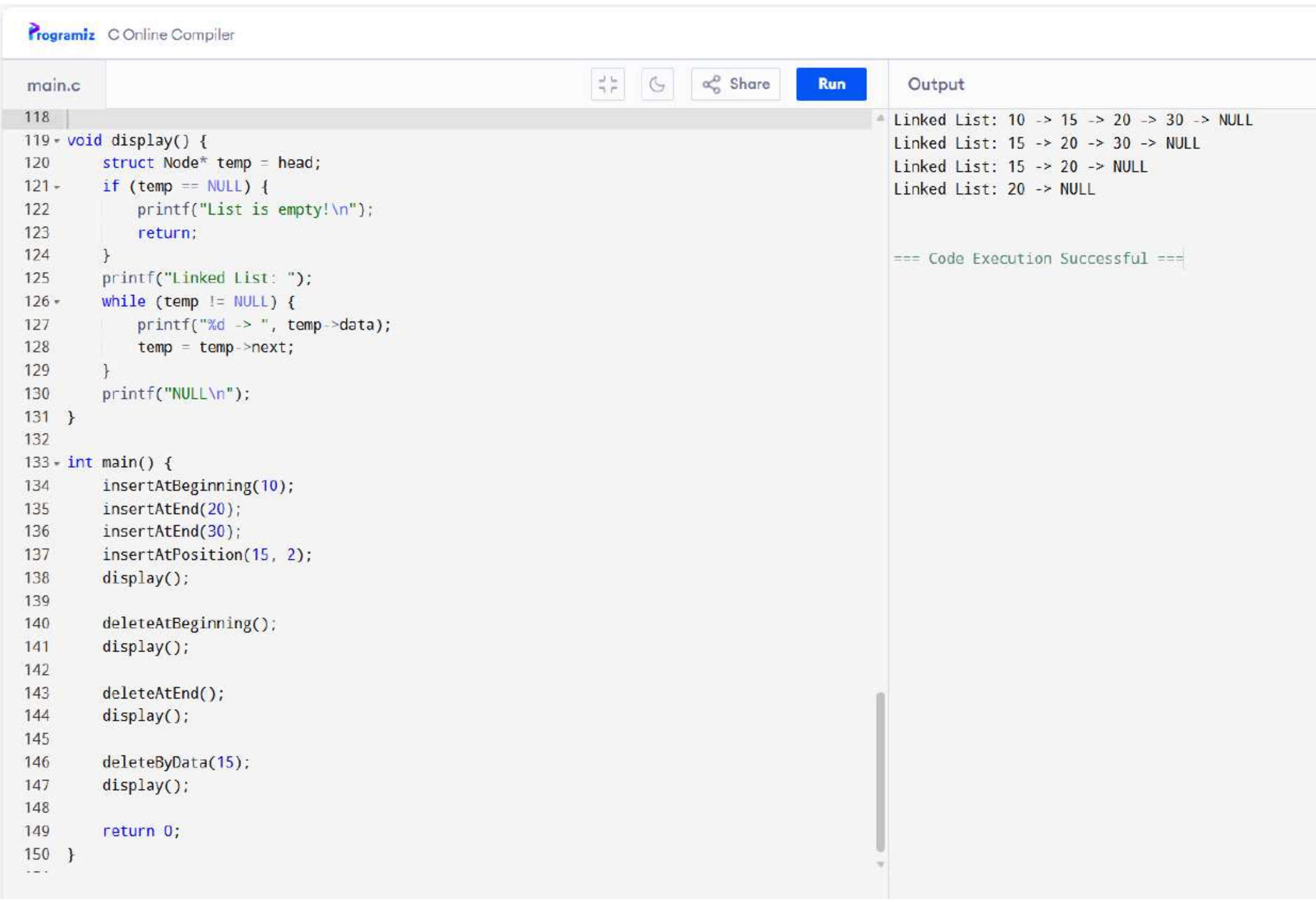
```c
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newNode;
}


void insertAtPosition(int data, int position) {
    if (position < 1) {
        printf("Invalid position!\n");
        return;
    }
    if (position == 1) {
        insertAtBeginning(data);
        return;
    }
    struct Node* newNode = createNode(data);
    struct Node* temp = head;
    for (int i = 1; temp != NULL && i < position - 1; i++) {
        temp = temp->next;
    }
    if (temp == NULL) {
        printf("Position out of range!\n");
        return;
    }
    newNode->next = temp->next;
    temp->next = newNode;
}


void deleteAtBeginning() {
    if (head == NULL) {
        printf("List is empty!\n");
        return;
```

```
Linked List: 10 -> 15 -> 20 -> 30 -> NULL
Linked List: 15 -> 20 -> 30 -> NULL
Linked List: 15 -> 20 -> NULL
Linked List: 20 -> NULL


=== Code Execution Successful ===
```



```c
void deleteAtEnd() {
    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }
    if (head->next == NULL) {
        free(head);
        head = NULL;
        return;
    }
    struct Node* temp = head;
    while (temp->next->next != NULL) {
        temp = temp->next;
    }
    free(temp->next);
    temp->next = NULL;
}


void deleteByData(int data) {
    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }
    if (head->data == data) {
        struct Node* temp = head;
        head = head->next;
        free(temp);
        return;
    }
    struct Node* temp = head;
    while (temp->next != NULL && temp->next->data != data) {
        temp = temp->next;
```

```
Linked List: 10 -> 15 -> 20 -> 30 -> NULL
Linked List: 15 -> 20 -> 30 -> NULL
Linked List: 15 -> 20 -> NULL
Linked List: 20 -> NULL


=== Code Execution Successful ===
```



```c

void display() {
    struct Node* temp = head;
    if (temp == NULL) {
        printf("List is empty!\n");
        return;
    }
    printf("Linked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    insertAtBeginning(10);
    insertAtEnd(20);
    insertAtEnd(30);
    insertAtPosition(15, 2);
    display();

    deleteAtBeginning();
    display();

    deleteAtEnd();
    display();

    deleteByData(15);
    display();

    return 0;
}
```

```
Linked List: 10 -> 15 -> 20 -> 30 -> NULL
Linked List: 15 -> 20 -> 30 -> NULL
Linked List: 15 -> 20 -> NULL
Linked List: 20 -> NULL


=== Code Execution Successful ===
```