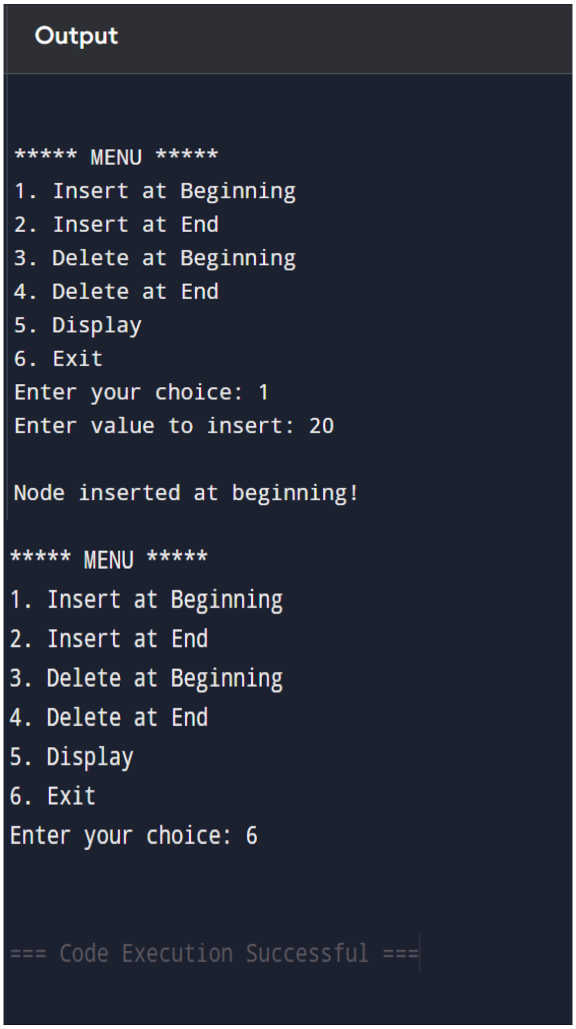
```c
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* next;
};
struct Node* head = NULL;
void insertAtBeginning(int);
void insertAtEnd(int);
void deleteAtBeginning();
void deleteAtEnd();
void display();
int main() {
    int choice, value;
    while (1) {
        printf("\n\n***** MENU *****\n");
        printf("1. Insert at Beginning\n");
        printf("2. Insert at End\n");
        printf("3. Delete at Beginning\n");
        printf("4. Delete at End\n");
        printf("5. Display\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch(choice) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &value);
                insertAtBeginning(value);
                break;
            case 2:
                printf("Enter value to insert: ");
                scanf("%d", &value);
                insertAtEnd(value);
                break;
            case 3:
                deleteAtBeginning();
                break;
            case 4:
                deleteAtEnd();
                break;
            case 5:
                display();
                break;
            case 6:
                exit(0);
            default:
                printf("\nInvalid choice! Try again.");
        }
    }
    return 0;
}
void insertAtBeginning(int value) {
    struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = head;
    head = newNode;
    printf("\nNode inserted at beginning!");
}
void insertAtEnd(int value) {
    struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));
    struct Node* temp = head;
    newNode->data = value;
    newNode->next = NULL;
    if (head == NULL) {
        head = newNode;
    } else {
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newNode;
    }
    printf("\nNode inserted at end!");
}
void deleteAtBeginning() {
    if (head == NULL) {
        printf("\nList is empty! Deletion not possible.");
    } else {
        struct Node* temp = head;
        printf("\nDeleted: %d", temp->data);
        head = head->next;
        free(temp);
    }
}
void deleteAtEnd() {
    if (head == NULL) {
        printf("\nList is empty! Deletion not possible.");
    } else if (head->next == NULL) {
        printf("\nDeleted: %d", head->data);
        free(head);
        head = NULL;
    } else {
        struct Node* temp = head;
        while (temp->next->next != NULL)
            temp = temp->next;
        printf("\nDeleted: %d", temp->next->data);
        free(temp->next);
        temp->next = NULL;
    }
}
void display() {
    if (head == NULL) {
        printf("\nList is empty!");
    } else {
        struct Node* temp = head;
        printf("\nLinked List elements: ");
        while (temp != NULL) {
            printf("%d -> ", temp->data);
            temp = temp->next;
        }
        printf("NULL");
    }
}
```

Output:



```
***** MENU *****
1. Insert at Beginning
2. Insert at End
3. Delete at Beginning
4. Delete at End
5. Display
6. Exit
Enter your choice: 1
Enter value to insert: 20

Node inserted at beginning!

***** MENU *****
1. Insert at Beginning
2. Insert at End
3. Delete at Beginning
4. Delete at End
5. Display
6. Exit
Enter your choice: 6


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