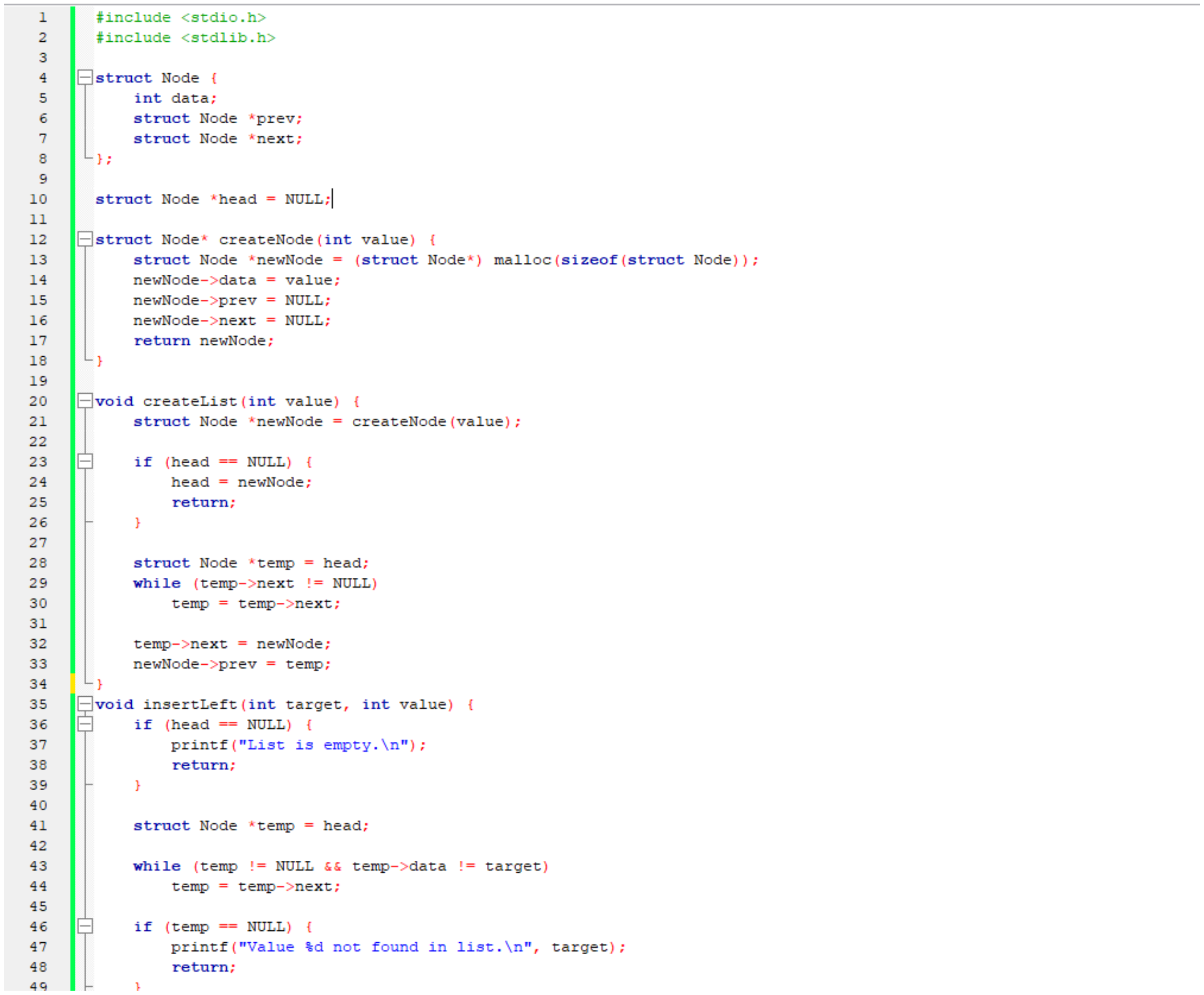
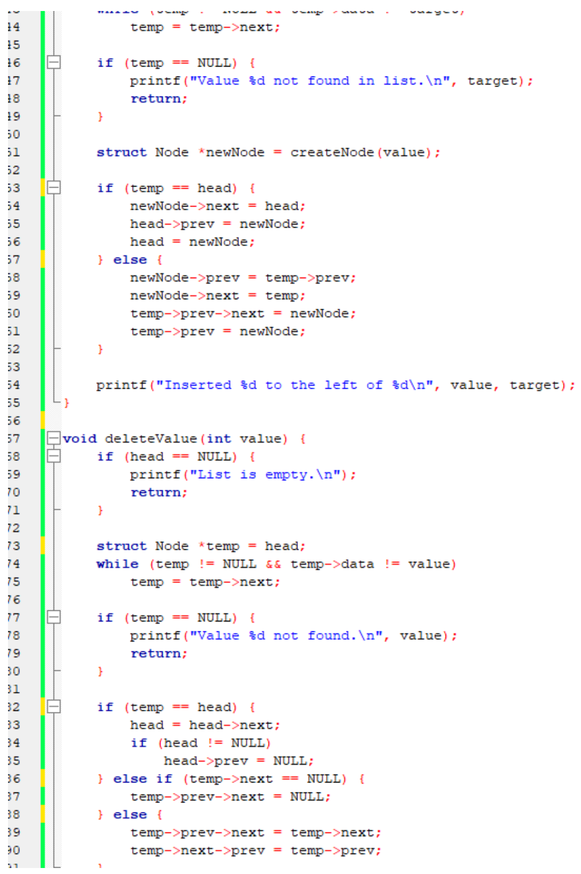
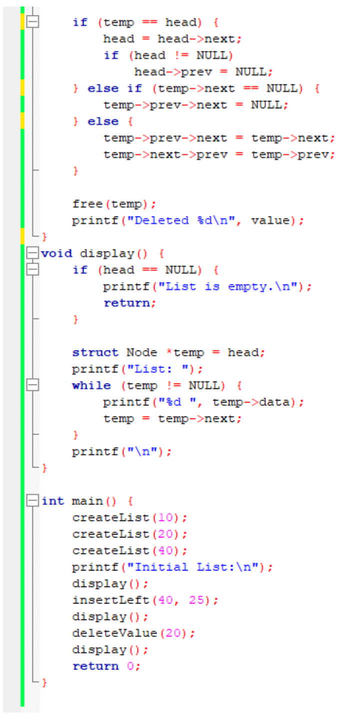


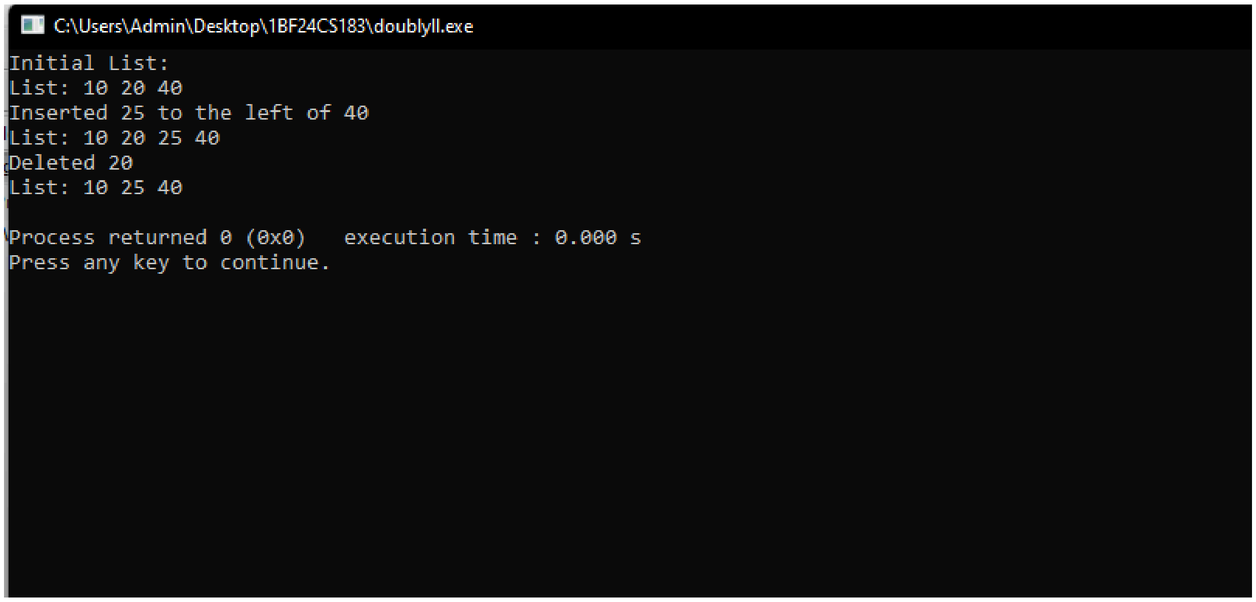
```c
#include <stdio.h>
#include <stdlib.h>

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

struct Node *head = NULL;

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

void createList(int value) {
    struct Node *newNode = createNode(value);

    if (head == NULL) {
        head = newNode;
        return;
    }

    struct Node *temp = head;
    while (temp->next != NULL)
        temp = temp->next;

    temp->next = newNode;
    newNode->prev = temp;
}
void insertLeft(int target, int value) {
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }

    struct Node *temp = head;

    while (temp != NULL && temp->data != target)
        temp = temp->next;

    if (temp == NULL) {
        printf("Value %d not found in list.\n", target);
        return;
    }
```



```c
    while (temp != NULL && temp->data != target)
        temp = temp->next;

    if (temp == NULL) {
        printf("Value %d not found in list.\n", target);
        return;
    }

    struct Node *newNode = createNode(value);

    if (temp == head) {
        newNode->next = head;
        head->prev = newNode;
        head = newNode;
    } else {
        newNode->prev = temp->prev;
        newNode->next = temp;
        temp->prev->next = newNode;
        temp->prev = newNode;
    }

    printf("Inserted %d to the left of %d\n", value, target);
}

void deleteValue(int value) {
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }

    struct Node *temp = head;
    while (temp != NULL && temp->data != value)
        temp = temp->next;

    if (temp == NULL) {
        printf("Value %d not found.\n", value);
        return;
    }

    if (temp == head) {
        head = head->next;
        if (head != NULL)
            head->prev = NULL;
    } else if (temp->next == NULL) {
        temp->prev->next = NULL;
    } else {
        temp->prev->next = temp->next;
        temp->next->prev = temp->prev;
    }
```



```c
    if (temp == head) {
        head = head->next;
        if (head != NULL)
            head->prev = NULL;
    } else if (temp->next == NULL) {
        temp->prev->next = NULL;
    } else {
        temp->prev->next = temp->next;
        temp->next->prev = temp->prev;
    }

    free(temp);
    printf("Deleted %d\n", value);
}
void display() {
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }

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

int main() {
    createList(10);
    createList(20);
    createList(40);
    printf("Initial List:\n");
    display();
    insertLeft(40, 25);
    display();
    deleteValue(20);
    display();
    return 0;
}
```



C:\Users\Admin\Desktop\1BF24CS183\doublyll.exe

```
Initial List:
List: 10 20 40
Inserted 25 to the left of 40
List: 10 20 25 40
Deleted 20
List: 10 25 40

Process returned 0 (0x0)   execution time : 0.000 s
Press any key to continue.
```