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

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

void insertEnd(struct Node** head, char data) {
    struct Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* temp = *head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newNode;
    }
}

struct Node* reverseList(struct Node* head) {
    struct Node *prev = NULL, *current = head, *next = NULL;
    while (current != NULL) {
        next = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
    return prev;
}

int isPalindrome(struct Node* head) {

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

```
    while (head != NULL && reversed != NULL) {
        if (head->data != reversed->data)
            return 0;
        head = head->next;
        reversed = reversed->next;
    }

    return 1;
}

int main() {
    char str[100];
    struct Node* head = NULL;

    printf("Enter a string: ");
    scanf("%s", str);

    for (int i = 0; i < strlen(str); i++) {
        insertEnd(&head, str[i]);
    }

    if (isPalindrome(head))
        printf("The string is a Palindrome.\n");
    else
        printf("The string is NOT a Palindrome.\n");

    return 0;
}
```