**Code**

// Author: Gokul Raj, 235, R3A, CSE

// QUESTION: Store string in doubly linked list. Check if palindrome.

#include <stdio.h>

#include <stdlib.h>

#define ARR\_LEN 256

// Structure to hold Doubly Linked list for string.

struct Node{

    char data;

    struct Node \*prev;

    struct Node \*next;

};

// Variables to hold HEAD, TAIL and LENGTH of string.

struct Node \*str\_head = NULL;

struct Node \*str\_tail = NULL;

int len = 1;

void loadString(){

    printf("Enter string: ");

    // Load first charecter to list

    struct Node \*firstNode = malloc(sizeof(struct Node));

    char c = getc(stdin);

    firstNode->data = c;

    firstNode->next = NULL;

    firstNode->prev = NULL;

    str\_tail = firstNode;

    str\_head = firstNode;

    c = getc(stdin);

    // Input charecters until end of input string

    // Set tail as the new node, increment length

    while(c != '\n'){

        struct Node \*newNode = malloc(sizeof(struct Node));

        newNode->data = c;

        newNode->next = NULL;

        newNode->prev = str\_tail;

        str\_tail->next = newNode;

        str\_tail = newNode;

        len++;

        c = getc(stdin);

    }

}

int checkPalindrome(struct Node \*head, struct Node \*tail, int len){

    struct Node \*f = head, \*r = tail;

    // Check chars from front and rear simultaneously, until reaching mid

    int i = 0;

    while(i < len/2){

        // if mismatched anywhere, not a palindrome

        if(f->data != r->data)

            return 0;

        f = f->next;

        r = r->prev;

        i++;

    }

    return 1;

}

// Driver code

void main(){

    loadString();

    if(checkPalindrome(str\_head, str\_tail, len))

        printf("Palindrome!!\n\n");

    else

        printf("Not Palindrome!!\n\n");

}

**Output**

Text

Description automatically generated