

# Rajalakshmi Engineering College

Name: JEYAGURU M

Email: 241801108@rajalakshmi.edu.in

Roll no:

Phone: 8072274688

Branch: REC

Department: I AI & DS FB

Batch: 2028

Degree: B.E - AI & DS

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 2\_COD\_Question 1

Attempt : 1

Total Mark : 10

Marks Obtained : 0

#### Section 1 : Coding

##### 1. Problem Statement

Your task is to create a program to manage a playlist of items. Each item is represented as a character, and you need to implement the following operations on the playlist.

Here are the main functionalities of the program:

Insert Item: The program should allow users to add items to the front and end of the playlist. Items are represented as characters. Display Playlist: The program should display the playlist containing the items that were added.

To implement this program, a doubly linked list data structure should be used, where each node contains an item character.

***Input Format***

The input consists of a sequence of space-separated characters, representing the items to be inserted into the doubly linked list.

The input is terminated by entering - (hyphen).

### ***Output Format***

The first line of output prints "Forward Playlist: " followed by the linked list after inserting the items at the end.

The second line prints "Backward Playlist: " followed by the linked list after inserting the items at the front.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: a b c -

Output: Forward Playlist: a b c

Backward Playlist: c b a

### ***Answer***

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

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

```
struct Node {  
    char item;  
    struct Node* next;  
    struct Node* prev;  
};
```

```
void insertAtEnd(struct Node** head, char item) {  
    struct Node* newNode = (Node*) malloc(sizeof(Node*));  
    if(*head==NULL){  
        *head=newNode;  
        return;  
    }  
    struct Node* temp = *head;  
    while(temp->next!=NULL){  
        temp=temp->next;  
    }  
}
```

```

    temp -> next = newNode;
    newNode -> prev = temp;
}
void displayForward(struct Node* head) {
    if(head==NULL){
        return;
    }
    printf("Forward Playlist: ");
    struct Node* temp=head;
    while(head!=NULL){
        printf("%c ",temp->item);
        temp=temp->next;
    }
    printf("\n");
}

```

```

void displayBackward(struct Node* tail) {
    if(tail==NULL){
        return;
    }
    struct Node* temp = tail;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    printf("Backwards Playlist: ");
    while(temp!=NULL){
        printf("%c ",temp->item);
        temp=temp->prev;
    }
    printf("\n");
}

```

```

void freePlaylist(struct Node* head) {
    struct Node* temp;
    while(head!=NULL){
        temp=head;
        head=head->next;
        free(temp);
    }
}

```

```

int main() {
    struct Node* playlist = NULL;

```

```
char item;

while (1) {
    scanf(" %c", &item);
    if (item == '-') {
        break;
    }
    insertAtEnd(&playlist, item);
}

struct Node* tail = playlist;
while (tail->next != NULL) {
    tail = tail->next;
}

printf("Forward Playlist: ");
displayForward(playlist);

printf("Backward Playlist: ");
displayBackward(tail);

freePlaylist(playlist);

return 0;
}
```

**Status : Wrong**

**Marks : 0/10**