

Rajalakshmi Engineering College

Name: Vithyavarshini P
Email: 240801387@rajalakshmi.edu.in
Roll no: 240801387
Phone: 9361308913
Branch: REC
Department: I ECE AF
Batch: 2028
Degree: B.E - ECE

Scan to verify results



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 1

Attempt : 2
Total Mark : 10
Marks Obtained : 10

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 = (struct Node*)malloc(sizeof(struct Node));  
    if(newNode == NULL){  
        printf("Memory allocation failed\n");  
        return;  
    }  
    newNode->item = item;  
    newNode->next = NULL;  
    if(*head == NULL) {
```

```

        newNode->prev = NULL;
        *head = newNode;
    }
    else{
        struct Node* current = *head;
        while(current->next != NULL){
            current = current->next;
        }
        newNode->prev = current;
        current->next = newNode;
    }
}

void displayForward(struct Node* head){
    struct Node* current = head;
    while(current!=NULL){
        printf("%c ",current->item);
        current = current->next;
    }
    printf("\n");
}

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

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

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 : Correct

Marks : 10/10