Rajalakshmi Engineering College

Name: Sanjay Kumar K
Email: 241801244@rajalakshmi.edu.in
Roll no: 241801244
Phone: 9710199820
Branch: REC
Department: I AI & DS FD
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ravi is developing a student registration system for a college. To efficiently store and manage the student IDs, he decides to implement a doubly linked list where each node represents a student's ID.

In this system, each student's ID is stored sequentially, and the system needs to display all registered student IDs in the order they were entered.

Implement a program that creates a doubly linked list, inserts student IDs, and displays them in the same order.

Input Format

The first line contains an integer N the number of student IDs.

The second line contains N space-separated integers representing the student IDs.

Output Format

The output should display the single line containing N space-separated integers representing the student IDs stored in the doubly linked list.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
10 20 30 40 50

Output: 10 20 30 40 50

Answer

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>

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

struct Node* createNode(int data) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   if (newNode == NULL) {
      fprintf(stderr, "Memory allocation failed\n");
      exit(EXIT_FAILURE);
   }
   newNode->data = data;
   newNode->next = NULL;
   newNode->prev = NULL;
   return newNode;
}

void insertAtEnd(struct Node** head, struct Node** tail, int data) {
   struct Node* newNode = createNode(data);
```

```
    if (*head == NULL) {
        *head = newNode;
        *tail = newNode;
    } else {
        newNode->prev = *tail;
        (*tail)->next = newNode;
        *tail = newNode;
    }
}

void displayList(struct Node* head) {
    struct Node* current = head;
    while (current != NULL) {
        printf("%d", current->data);
        current = current->next;
        if (current != NULL) {
            printf(" ");
        }
    }
    printf("\n");
}

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

int main() {
    int n, studentID;
    struct Node* head = NULL;
    struct Node* tail = NULL;

    if (scanf("%d", &n) != 1) {
        fprintf(stderr, "Error reading the number of student IDs\n");
        return 1;
    }
    for (int i = 0; i < n; i++) {
        if (scanf("%d", &studentID) != 1) {
            fprintf(stderr, "Error reading student ID\n");
            freeList(head);
            return 1;
        }
        insertAtEnd(&head, &tail, studentID);
    }

    displayList(head);

    freeList(head);
    return 0;
}
```

Status : Correct ***Marks : 10/10***