Rajalakshmi Engineering College

Name: Jaidev Arunachalam
Email: 241801099@rajalakshmi.edu.in
Roll no: 241801099
Phone: 9940254113
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 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

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

// Define a doubly linked list node
struct Node {
   int studentID;        // To store student ID
   struct Node* prev;    // Pointer to previous node
   struct Node* next;    // Pointer to next node
};

// Function to insert a new node at the end of the doubly linked list
void insertAtEnd(struct Node** head, int studentID) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->studentID = studentID;
   newNode->next = NULL;
   newNode->prev = NULL;

   if (*head == NULL) {
      *head = newNode; // If the list is empty, make the new node the head
   } else {
      struct Node* temp = *head;
      while (temp->next != NULL) {
         temp = temp->next;  // Traverse to the last node
      }
```

```
        temp->next = newNode;  // Link the new node to the last node
        newNode->prev = temp;  // Set the previous pointer of the new node
    }
}

// Function to display the student IDs stored in the doubly linked list
void displayList(struct Node* head) {
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d ", temp->studentID);  // Print the student ID
        temp = temp->next;               // Move to the next node
    }
    printf("\n");
}

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

    // Read the number of student IDs
    scanf("%d", &N);

    // Read the student IDs and insert them into the doubly linked list
    for (int i = 0; i < N; i++) {
        scanf("%d", &studentID);
        insertAtEnd(&head, studentID);
    }

    // Display the student IDs in the list
    displayList(head);

    return 0;
}
```

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