Rajalakshmi Engineering College

Name: Shremathi K
Email: 240701504@rajalakshmi.edu.in
Roll no:
Phone: 8870649491
Branch: REC
Department: I CSE FE
Batch: 2028
Degree: B.E - CSE

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 <iostream>
using namespace std;

// Node structure for the doubly linked list
struct Node {
   int studentID;
   Node* prev;
   Node* next;
};

// Function to insert at the end of the list
void insertAtEnd(Node*& head, Node*& tail, int id) {
   Node* newNode = new Node;
   newNode->studentID = id;
   newNode->prev = nullptr;
   newNode->next = nullptr;

   if (head == nullptr) {
      // First node in the list
      head = tail = newNode;
   } else {
      // Add at the end
      tail->next = newNode;
      newNode->prev = tail;
```

```
        tail = newNode;
    }
}

// Function to display the list
void displayList(Node* head) {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->studentID << " ";
        temp = temp->next;
    }
    cout << endl;
}

// Main function
int main() {
    int N;
    cin >> N;

    Node* head = nullptr;
    Node* tail = nullptr;

    for (int i = 0; i < N; ++i) {
        int id;
        cin >> id;
        insertAtEnd(head, tail, id);
    }

    displayList(head);

    return 0;
}
```

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