Rajalakshmi Engineering College

Name: Harshavarthini
Email: 240701181@rajalakshmi.edu.in
Roll no: 240701181
Phone: 9150394958
Branch: REC
Department: I CSE AH
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

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

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

typedef struct DoublyLinkedList {
    Node* head;
    Node* tail;
} DoublyLinkedList;

void append(DoublyLinkedList* list, int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->prev = NULL;
    newNode->next = NULL;

    if (list->tail == NULL) {  // If the list is empty
        list->head = list->tail = newNode;
    } else {
```

```
        list->tail->next = newNode;
        newNode->prev = list->tail;
        list->tail = newNode;
    }
}

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

int main() {
    DoublyLinkedList list = {NULL, NULL};

    int n, id;
    scanf("%d", &n);

    if (n < 1 || n > 10) {
        printf("Invalid input size!\n");
        return 0;
    }

    for (int i = 0; i < n; i++) {
        scanf("%d", &id);
        if (id < 1 || id > 1000000) {
            printf("Invalid ID!\n");
            return 0;
        }
        append(&list, id);
    }

    display(&list);

    return 0;
}
```

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