Rajalakshmi Engineering College

Name: arul santhosh X
Email: 241501022@rajalakshmi.edu.in
Roll no: 241501022
Phone: 9361955774
Branch: REC
Department: I AIML AD
Batch: 2028
Degree: B.E - AI & ML

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>


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


void append(struct node** head, int data) {
    struct node* newNode = (struct node*)malloc(sizeof(struct node));
    newNode->data = data;
    newNode->next = NULL;
    newNode->prev = NULL;

    if (*head == NULL) {
        *head = newNode;
        return;
    }

    struct node* temp = *head;
    while (temp->next != NULL)
        temp = temp->next;

    temp->next = newNode;
    newNode->prev = temp;
}


void rotateClockwise(struct node** head, int k) {
    if (*head == NULL || k == 0)
        return;

    struct node* last = *head;
    int len = 1;


    while (last->next != NULL) {
        last = last->next;
        len++;
    }

    k = k % len;
    if (k == 0)
        return;


    struct node* newTail = *head;
    for (int i = 0; i < len - k - 1; i++)
        newTail = newTail->next;

    struct node* newHead = newTail->next;


    newTail->next = NULL;
    newHead->prev = NULL;

    last->next = *head;
    (*head)->prev = last;

    *head = newHead;
}
void printList(struct node* head) {
    while (head != NULL) {
        printf("%d ", head->data);
        head = head->next;
    }
    printf("\n");
}

int main() {
    int n, k, val;
    struct node* head = NULL;

    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &val);
        append(&head, val);
    }

    scanf("%d", &k);

    rotateClockwise(&head, k);
    printList(head);

    return 0;
}
```

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