

Rajalakshmi Engineering College

Name: PRAVEEN P
Email: 240801249@rajalakshmi.edu.in
Roll no: 240801249
Phone: 8608588599
Branch: REC
Department: I ECE AF
Batch: 2028
Degree: B.E - ECE

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 id;
```

```
    struct Node* next;
```

```
    struct Node* prev;
```

```
}Node;
```

```
typedef struct DoublyLinkedList  
{
```

```
    Node* head;
```

```
}DoublyLinkedList;
```

```
void init_list(DoublyLinkedList* list)
```

```
{
```

```
    list->head = NULL;
```

```
}
```

```
void insert_end(DoublyLinkedList* list,int id)
```

```
{
```

```
    Node* new_node = (Node*)malloc(sizeof(Node));
```

```
    new_node->id = id;
```

```

new_node->next = NULL;
if(list->head == NULL)
{
    new_node->prev = NULL;
    list->head = new_node;
}
else
{
    Node* temp = list->head;
    while(temp->next != NULL)
    {
        temp = temp->next;
    }
    temp->next = new_node;
    new_node->prev = temp;
}
}

```

```

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

```

```

int main()
{
    int n;
    scanf("%d",&n);

    DoublyLinkedList list;
    init_list(&list);

    for(int i=0;i<n;i++)
    {
        int id;scanf("%d",&id);
        insert_end(&list, id);
    }
}

```

```
}  
print_list(&list);  
return 0;  
}
```

Status : Correct

Marks : 10/10