# Rajalakshmi Engineering College

Name: KISHORE  RAJ.A.S.
Email: 241801127@rajalakshmi.edu.in
Roll no: 241801127
Phone: 7397295789
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***

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>
struct Node{
    int studentid;
    struct Node*prev;
    struct Node*next;
};
struct dll{
    struct Node*head;
    struct Node*tail;
};
struct Node*createNode(int studentid){
    struct Node*nn=(struct Node*)malloc(sizeof(struct Node));
    if(nn==NULL){
        printf("Memory allocation failed\n");
        exit(1);
    }
    nn->studentid=studentid;
    nn->prev=NULL;
    nn->next=NULL;
    return nn;
}
struct dll*createdll(){
```

```
    struct dll*list=(struct dll*)malloc(sizeof(struct dll));
    if(list==NULL){
    printf("Memory allocation failed\n");
    exit(1);
    }
    list->head=NULL;
    list->tail=NULL;
    return list;
}
void append(struct dll*list,int studentid){
    struct Node*nn=createNode(studentid);
    if(list->head==NULL){
        list->head=nn;
        list->tail=nn;
    }
    else{
        list->tail->next=nn;
        nn->prev=list->tail;
        list->tail=nn;
    }
}
void dl(struct dll*list){
    struct Node*current=list->head;
    while(current!=NULL){
        printf("%d ",current->studentid);
        current=current->next;
    }
    printf("\n");
}
int main(){
    int numstudents;
    scanf("%d",&numstudents);
    struct dll*studentlist=createdll();
    for(int i=0;i<numstudents;i++){
        int studentid;
        scanf("%d",&studentid);
        append(studentlist,studentid);
    }
    dl(studentlist);
    struct Node*current=studentlist->head;
    while(current!=NULL){
        struct Node*temp=current;
```

```
        current=current->next;
        free(temp);
    }
    free(studentlist);
    return 0;
}
```

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