# Rajalakshmi Engineering College

Name: Navaneetha Krishnan
Email: 241901067@rajalakshmi.edu.in
Roll no: 241901067
Phone: 8939010233
Branch: REC
Department: l CSE (CS) FB
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

## NeoColab_REC_CS23231_DATA STRUCTURES

### REC_DS using C_Week 2_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 30

## Section 1 : Coding

1.  Problem Statement

You are required to implement a program that deals with a doubly linked list.

The program should allow users to perform the following operations:

Insertion at the End: Insert a node with a given integer data at the end of the doubly linked list.Insertion at a given Position: Insert a node with a given integer data at a specified position within the doubly linked list.Display the List: Display the elements of the doubly linked list.

*Input Format*

The first line of input consists of an integer n, representing the number of elements to be initially inserted into the doubly linked list.

The second line consists of n space-separated integers, denoting the elements to be inserted at the end.

The third line consists of integer m, representing the new element to be inserted.

The fourth line consists of an integer p, representing the position at which the new element should be inserted (1-based indexing).

*Output Format*

If p is valid, display the elements of the doubly linked list after performing the insertion at the specified position.

If p is invalid, display "Invalid position" in the first line and the second line prints the original list.

Refer to the sample output for formatting specifications.

*Sample Test Case*

Input: 5
10 25 34 48 57
35
4
Output: 10 25 34 35 48 57

*Answer*

```
// You are using GCC
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node*prev,*next;
}*head=NULL,*tail=NULL;
void create(int data){
    struct node*Newnode=(struct node*)malloc(sizeof(struct node));
    Newnode->data=data;
    Newnode->next=NULL;
    if(head==NULL){
        Newnode->prev=NULL;
        head=tail=Newnode;
```

```c
    }
    else{
        tail->next=Newnode;
        Newnode->prev=tail;
        tail=Newnode;
    }
}
void InsertAtPosition(int a,int pos){
    struct node*Newnode=(struct node*)malloc(sizeof(struct node));
    struct node*temp=head;
    int i=1;
    Newnode->data=a;
    Newnode->next=NULL;
    Newnode->prev=NULL;
    if(pos==1){
        Newnode->next=temp;
        temp->prev=Newnode;
    }
    else{
    while(i<pos-1){
        temp=temp->next;
        i++;
    }
    Newnode->next=temp->next;
    Newnode->prev=temp;
     temp->next->prev=Newnode;
     temp->next=Newnode;
    }
}
void display(){
    struct node*temp=head;
    while(temp!=NULL){
        printf("%d ",temp->data);
        temp=temp->next;
    }
}
int main(){
    int n,a,b,pos;
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d ",&a);
        create(a);
```

```
    }
    scanf("%d\n%d",&b,&pos);
    if(pos<0 || pos>n)
    printf("Invalid position\n");
    else
     InsertAtPosition(b,pos);
    display();
}
```

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

2.  Problem Statement

Aarav is working on a program to analyze his test scores, which are stored
in a doubly linked list. He needs a solution to input scores into the list and
determine the highest score.

Help him by providing code that lets users enter test scores into the doubly
linked list and find the maximum score efficiently.

*Input Format*

The first line consists of an integer N, representing the number of elements to be
initially inserted into the doubly linked list.

The second line consists of N space-separated integers, denoting the score to
be inserted.

*Output Format*

The output prints an integer, representing the highest score present in the list.

Refer to the sample output for formatting specifications.

*Sample Test Case*

Input: 4
89 71 2 70
Output: 89

*Answer*

```c
// You are using GCC
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node*prev,*next;
}*head=NULL,*tail=NULL;
void create(int a){
    struct node*Newnode=(struct node*)malloc(sizeof(struct node));
    Newnode->data=a;
    Newnode->prev=NULL;
    Newnode->next=NULL;
    if(head==NULL){
        head=tail=Newnode;
    }
    else{
        tail->next=Newnode;
        Newnode->prev=tail;
        tail=Newnode;
    }
}
void display(){
    struct node*temp=head;
    int max=0;
    while(temp!=NULL){
        if(temp->data >max){
            max=temp->data;
        }
        temp=temp->next;
    }
    printf("%d",max);
}
int main(){
    int n,a;
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d ",&a);
        create(a);
    }
    display();
    return 0;
```

}

3.   Problem Statement

Krishna needs to create a doubly linked list to store and display a
sequence of integers. Your task is to help write a program to read a list of
integers from input, store them in a doubly linked list, and then display the
list.

*Input Format*

The first line of input consists of an integer n, representing the number of
integers in the list.

The second line of input consists of n space-separated integers.

*Output Format*

The output prints a single line displaying the integers in the order they were
added to the doubly linked list, separated by spaces.

If nothing is added (i.e., the list is empty), it will display "List is empty".

Refer to the sample output for the formatting specifications.

*Sample Test Case*

Input: 5
1 2 3 4 5
Output: 1 2 3 4 5

*Answer*

```
// You are using GCC
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node*prev,*next;
```

```c
}*head=NULL,*tail=NULL;
void create(int a){
    struct node*Newnode=(struct node*)malloc(sizeof(struct node));
    Newnode->data=a;
    Newnode->prev=NULL;
    Newnode->next=NULL;
    if(head==NULL){
        head=tail=Newnode;
    }
    else{
        tail->next=Newnode;
        Newnode->prev=tail;
        tail=Newnode;
    }
}
void display(){
    struct node*temp=head;
    if(head==NULL){
        printf("List is empty");
    }
    else{
    while(temp!=NULL){
        printf("%d ",temp->data);
        temp=temp->next;
    }
    }
}
int main(){
    int n,a;
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d ",&a);
        create(a);
    }
    if(n==0)
    printf("List is empty");
    else
    display();
    return 0;
}
```

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