

# Rajalakshmi Engineering College

Name: Pratiba S  
Email: 241901080@rajalakshmi.edu.in  
Roll no: 241901080  
Phone: 7200010194  
Branch: REC  
Department: I 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\_COD\_Question 1

Attempt : 1  
Total Mark : 10  
Marks Obtained : 0

#### Section 1 : Coding

##### 1. Problem Statement

Your task is to create a program to manage a playlist of items. Each item is represented as a character, and you need to implement the following operations on the playlist.

Here are the main functionalities of the program:

Insert Item: The program should allow users to add items to the front and end of the playlist. Items are represented as characters. Display Playlist: The program should display the playlist containing the items that were added.

To implement this program, a doubly linked list data structure should be used, where each node contains an item character.

***Input Format***

The input consists of a sequence of space-separated characters, representing the items to be inserted into the doubly linked list.

The input is terminated by entering - (hyphen).

### ***Output Format***

The first line of output prints "Forward Playlist: " followed by the linked list after inserting the items at the end.

The second line prints "Backward Playlist: " followed by the linked list after inserting the items at the front.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: a b c -

Output: Forward Playlist: a b c

Backward Playlist: c b a

### ***Answer***

-

**Status :** Skipped

**Marks :** 0/10

# Rajalakshmi Engineering College

Name: Pratiba S  
Email: 241901080@rajalakshmi.edu.in  
Roll no: 241901080  
Phone: 7200010194  
Branch: REC  
Department: I 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\_COD\_Question 2

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Moniksha, a chess coach organizing a tournament, needs a program to manage participant IDs efficiently. The program maintains a doubly linked list of IDs and offers two functions: Append to add IDs as students register, and Print Maximum ID to identify the highest ID for administrative tasks.

This tool streamlines tournament organization, allowing Moniksha to focus on coaching her students effectively.

##### ***Input Format***

The first line consists of an integer  $n$ , representing the number of participant IDs to be added.

The second line consists of  $n$  space-separated integers representing the participant IDs.

### **Output Format**

The output displays a single integer, representing the maximum participant ID.

If the list is empty, the output prints "Empty list!".

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 3

163 137 155

Output: 163

### **Answer**

```
#include <stdio.h>
#include <stdlib.h>
struct Node{
    int data;
    struct Node* prev;
    struct Node* next;
};
struct Node* create(int data){
    struct Node* newnode=(struct Node*)malloc(sizeof(struct Node));
    newnode->data=data;
    newnode->next=NULL;
    return newnode;
}
void append(struct Node** headref,int data){
    struct Node* newnode=create(data);
    if(*headref==NULL){
        *headref=newnode;
        return;
    }
    struct Node* temp=*headref;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    temp->next=newnode;
    newnode->prev=temp;
}
```

```

}
void print(struct Node* head){
    if(head==NULL){
        printf("Empty list!\n");
        return;
    }
    int max=head->data;
    struct Node*temp=head->next;
    while(temp!=NULL){
        if(temp->data>max){
            max=temp->data;
        }
        temp=temp->next;
    }
    printf("%d\n",max);
}
int main(){
    int n,id;
    struct Node* head=NULL;
    if(scanf("%d",&n)!=1 || n<0){
        printf("Invalid input!\n");
        return 1;
    }
    for(int i=0;i<n;i++){
        if(scanf("%d",&id)!=1){
            printf("Invalid input\n");
            return 1;
        }
        append(&head,id);
    }
    print(head);
}

```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Pratiba S  
Email: 241901080@rajalakshmi.edu.in  
Roll no: 241901080  
Phone: 7200010194  
Branch: REC  
Department: I 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\_COD\_Question 3

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Bob is tasked with developing a company's employee record management system. The system needs to maintain a list of employee records using a doubly linked list. Each employee is represented by a unique integer ID.

Help Bob to complete a program that adds employee records at the front, traverses the list, and prints the same for each addition of employees to the list.

##### ***Input Format***

The first line of input consists of an integer N, representing the number of employees.

The second line consists of N space-separated integers, representing the employee IDs.

### **Output Format**

For each employee ID, the program prints "Node Inserted" followed by the current state of the doubly linked list in the next line, with the data values of each node separated by spaces.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 4

101 102 103 104

Output: Node Inserted

101

Node Inserted

102 101

Node Inserted

103 102 101

Node Inserted

104 103 102 101

### **Answer**

```
#include <iostream>
using namespace std;
```

```
struct node {
    int info;
    struct node* prev, * next;
};
```

```
struct node* start = NULL;
```

```
// You are using GCC
```

```
struct node* createnode(int value){
    struct node* newnode=(struct node*)malloc(sizeof(struct node));
    newnode->info=value;
    newnode->prev=NULL;
    newnode->next=NULL;
    return newnode;
}
```

```
void traverse() {
```

```

    struct node*temp=start;
    printf("Node inserted\n");
    while(temp!=NULL){
        printf("%d",temp->info);
        temp=temp->next;
    }
    printf("\n");
}

void insertAtFront(int data) {
    struct node*newnode=createnode(data);
    if(start!=NULL){
        newnode->next=start;
        start->prev=newnode;
    }
    start=newnode;
}

int main() {
    int n, data;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> data;
        insertAtFront(data);
        traverse();
    }
    return 0;
}

```

**Status :** Correct

**Marks :** 10/10



# Rajalakshmi Engineering College

Name: Pratiba S  
Email: 241901080@rajalakshmi.edu.in  
Roll no: 241901080  
Phone: 7200010194  
Branch: REC  
Department: I 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\_COD\_Question 4

Attempt : 1  
Total Mark : 10  
Marks Obtained : 0

#### 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**

-

**Status :** Skipped

**Marks :** 0/10

# Rajalakshmi Engineering College

Name: Pratiba S  
Email: 241901080@rajalakshmi.edu.in  
Roll no: 241901080  
Phone: 7200010194  
Branch: REC  
Department: I 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

Imagine Anu is tasked with finding the middle element of a doubly linked list. Given a doubly linked list where each node contains an integer value and is inserted at the end, implement a program to find the middle element of the list. If the number of nodes is even, return the middle element pair.

#### ***Input Format***

The first line of input consists of an integer N, representing the number of nodes in the doubly linked list.

The second line consists of N space-separated integers, representing the values of the nodes in the doubly linked list.

#### ***Output Format***

The first line of output prints the space-separated elements of the doubly linked list.

The second line prints the middle element(s) of the doubly linked list, depending on whether the number of nodes is odd or even.

Refer to the sample outputs for the formatting specifications.

### **Sample Test Case**

Input: 5

10 20 30 40 50

Output: 10 20 30 40 50

30

### **Answer**

```
#include <stdio.h>
#include<stdlib.h>
struct Node{
    int value;
    struct Node* prev;
    struct Node* next;
};
struct Node*createNode(int value){
    struct Node* newNode=(struct Node*)malloc(sizeof(struct Node));
    newNode->value=value;
    newNode->prev=NULL;
    newNode->next=NULL;
    return newNode;
}
void appendNode(struct Node** head,int value){
    struct Node* newNode=createNode(value);
    if(*head==NULL){
        *head=newNode;
        return;
    }
    struct Node* temp=*head;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    temp->next=newNode;
    newNode->prev=temp;
}
```

```

}
void printList(struct Node* head){
    struct Node* temp=head;
    while(temp!=NULL){
        printf("%d ",temp->value);
        temp=temp->next;
    }
    printf("\n");
}
void findMiddle(struct Node* head,int N){
    struct Node* slow=head;
    struct Node* fast=head;
    while(fast!=NULL && fast->next!=NULL){
        slow=slow->next;
        fast=fast->next->next;
    }
    if(fast!=NULL){
        printf("%d \n", slow->value);
    }else{
        printf("%d %d\n", slow->prev->value,slow->value);
    }
}
int main(){
    int N;
    scanf("%d", &N);
    int values[N];
    for(int i=0;i<N;i++){
        scanf("%d", &values[i]);
    }
    struct Node* head=NULL;
    for(int i=0;i<N;i++){
        appendNode(&head,values[i]);
    }
    printList(head);
    findMiddle(head,N);
    return 0;
}

```

**Status :** Correct

**Marks :** 10/10

## 2. Problem Statement

Vanessa is learning about the doubly linked list data structure and is eager to play around with it. She decides to find out how the elements are inserted at the beginning and end of the list.

Help her implement a program for the same.

### ***Input Format***

The first line of input contains an integer N, representing the size of the doubly linked list.

The next line contains N space-separated integers, each representing the values to be inserted into the doubly linked list.

### ***Output Format***

The first line of output prints the integers, after inserting them at the beginning, separated by space.

The second line prints the integers, after inserting at the end, separated by space.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 5

1 2 3 4 5

Output: 5 4 3 2 1

1 2 3 4 5

### ***Answer***

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* prev;
    struct Node* next;
};
void insertAtBeginning(struct Node** head, int data) {
```

```

    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->prev = NULL;
    newNode->next = *head;
    if (*head != NULL) {
        (*head)->prev = newNode;
    }
    *head = newNode;
}

void insertAtEnd(struct Node** head, int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    struct Node* temp = *head;
    newNode->data = data;
    newNode->next = NULL;
    if (*head == NULL) {
        newNode->prev = NULL;
        *head = newNode;
        return;
    }
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newNode;
    newNode->prev = temp;
}

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

int main() {
    int N, i;
    scanf("%d", &N);
    int elements[N];
    for (i = 0; i < N; i++) {
        scanf("%d", &elements[i]);
    }
    struct Node* head = NULL;
    for (i = 0; i < N; i++) {

```

```
        insertAtBeginning(&head, elements[i]);
    }
    printList(head);
    head = NULL;
    for (i = 0; i < N; i++) {
        insertAtEnd(&head, elements[i]);
    }
    printList(head);
    return 0;
}
```

**Status :** Correct

**Marks :** 10/10

### 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**

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* prev;
    struct Node* next;
};
void insertAtEnd(struct Node** head, int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->prev = NULL;
    newNode->next = NULL;
    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* temp = *head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newNode;
        newNode->prev = temp;
    }
}
void printList(struct Node* head) {
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d", temp->data);
        if (temp->next != NULL) {
            printf(" ");
        }
        temp = temp->next;
    }
    printf("\n");
}
int main(){
```

```
int n;  
scanf("%d", &n);  
if (n == 0) {  
    printf("List is empty\n");  
    return 0;  
}  
struct Node* head = NULL;  
for (int i = 0; i < n; i++) {  
    int value;  
    scanf("%d", &value);  
    insertAtEnd(&head, value);  
}  
printList(head);  
return 0;  
}
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Pratiba S  
Email: 241901080@rajalakshmi.edu.in  
Roll no: 241901080  
Phone: 7200010194  
Branch: REC  
Department: I 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\_PAH

Attempt : 1  
Total Mark : 50  
Marks Obtained : 0

### Section 1 : Coding

#### 1. Problem Statement

Riya is developing a contact management system where recently added contacts should appear first. She decides to use a doubly linked list to store contact IDs in the order they are added. Initially, new contacts are inserted at the front of the list. However, sometimes she needs to insert a new contact at a specific position in the list based on priority.

Help Riya implement this system by performing the following operations:

Insert contact IDs at the front of the list as they are added. Insert a new contact at a given position in the list.

#### ***Input Format***

The first line of input consists of an integer N, representing the initial size of the linked list.

The second line consists of N space-separated integers, representing the values of the linked list to be inserted at the front.

The third line consists of an integer position, representing the position at which the new value should be inserted (position starts from 1).

The fourth line consists of integer data, representing the new value to be inserted.

### ***Output Format***

The first line of output prints the original list after inserting initial elements to the front.

The second line prints the updated linked list after inserting the element at the specified position.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 4

10 20 30 40

3

25

Output: 40 30 20 10

40 30 25 20 10

### ***Answer***

-

**Status :** Skipped

**Marks :** 0/10

## **2. Problem Statement**

Bala is a student learning about the doubly linked list and its functionalities. He came across a problem where he wanted to create a doubly linked list by appending elements to the front of the list.

After populating the list, he wanted to delete the node at the given position from the beginning. Write a suitable code to help Bala.

**Input Format**

The first line contains an integer N, the number of elements in the doubly linked list.

The second line contains N integers separated by a space, the data values of the nodes in the doubly linked list.

The third line contains an integer X, the position of the node to be deleted from the doubly linked list.

**Output Format**

The first line of output displays the original elements of the doubly linked list, separated by a space.

The second line prints the updated list after deleting the node at the given position X from the beginning.

Refer to the sample output for formatting specifications.

**Sample Test Case**

Input: 5

10 20 30 40 50

2

Output: 50 40 30 20 10

50 30 20 10

**Answer**

-

**Status :** Skipped

**Marks :** 0/10

**3. Problem Statement**

Rohan is a software developer who is working on an application that

processes data stored in a Doubly Linked List. He needs to implement a feature that finds and prints the middle element(s) of the list. If the list contains an odd number of elements, the middle element should be printed. If the list contains an even number of elements, the two middle elements should be printed.

Help Rohan by writing a program that reads a list of numbers, prints the list, and then prints the middle element(s) based on the number of elements in the list.

### ***Input Format***

The first line of the input consists of an integer  $n$  the number of elements in the doubly linked list.

The second line consists of  $n$  space-separated integers representing the elements of the list.

### ***Output Format***

The first line prints the elements of the list separated by space. (There is an extra space at the end of this line.)

The second line prints the middle element(s) based on the number of elements.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 5

20 52 40 16 18

Output: 20 52 40 16 18

40

### ***Answer***

-

**Status :** Skipped

**Marks :** 0/10

## **4. Problem Statement**

Tom is a software developer working on a project where he has to check if a doubly linked list is a palindrome. He needs to write a program to solve this problem. Write a program to help Tom check if a given doubly linked list is a palindrome or not.

**Input Format**

The first line consists of an integer N, representing the number of elements in the linked list.

The second line consists of N space-separated integers representing the linked list elements.

**Output Format**

The first line displays the space-separated integers, representing the doubly linked list.

The second line displays one of the following:

1. If the doubly linked list is a palindrome, print "The doubly linked list is a palindrome".
2. If the doubly linked list is not a palindrome, print "The doubly linked list is not a palindrome".

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: 5

1 2 3 2 1

Output: 1 2 3 2 1

The doubly linked list is a palindrome

**Answer**

-

**Status :** Skipped

**Marks :** 0/10

Pranav wants to clockwise rotate a doubly linked list by a specified number of positions. He needs your help to implement a program to achieve this. Given a doubly linked list and an integer representing the number of positions to rotate, write a program to rotate the list clockwise.

***Input Format***

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

The second line consists of n space-separated linked list elements.

The third line consists of an integer k, representing the number of places to rotate the list.

***Output Format***

The output displays the elements of the doubly linked list after rotating it by k positions.

Refer to the sample output for the formatting specifications.

***Sample Test Case***

Input: 5  
1 2 3 4 5  
1

Output: 5 1 2 3 4

***Answer***

-

**Status :** Skipped

**Marks :** 0/10