Rajalakshmi Engineering College

Name: Shivani R J
Email: 240701500@rajalakshmi.edu.in
Roll no: 2116240701500
Phone: 9962571492
Branch: REC
Department: I CSE FE
Batch: 2028
Degree: B.E - CSE

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

Sam is learning about two-way linked lists. He came across a problem where he had to populate a two-way linked list and print the original as well as the reverse order of the list. Assist him with a suitable program.

Input Format

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

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

Output Format

The first line displays the message: "List in original order:"

The second line displays the elements of the doubly linked list in the original order.

The third line displays the message: "List in reverse order:"

The fourth line displays the elements of the doubly linked list in reverse order.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5
1 2 3 4 5
Output: List in original order:
1 2 3 4 5
List in reverse order:
5 4 3 2 1

Answer

```c
// You are using GCC
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node*prev;
    struct node*next;
};
void insertend(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;
        return;
    }
    struct node*temp=*head;
```

```c
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=newnode;
        newnode->prev=temp;
}
void print(struct node*head)
{
    struct node* temp =head;
    while(temp!=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->next;
    }
    printf("\n");
}
void printrev(struct node*head)
{
    struct node*temp=head;
    while(temp->next!=NULL)
    {
        temp=temp->next;
    }
    while(temp!=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->prev;
    }
    printf("\n");
}
int main()
{
    int n,i,val;
    struct node*head=NULL;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&val);
        insertend(&head,val);
    }
    printf("List in original order:\n");
```

```
    print(head);
    printf("List in reverse order:\n");
    printrev(head);
}
```

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

```
// You are using GCC
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node*next;
    struct node*prev;
};
void insertbeg(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 insertend(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)
    {
        newnode->prev=NULL;
        *head=newnode;
        return;
    }
    struct node*temp=*head;
    while(temp->next!=NULL)
    {
        temp=temp->next;
    }
    temp->next=newnode;
    newnode->prev=temp;
}
void print(struct node*head)
{
```

```c
    struct node*temp=head;
    while(temp!=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->next;
    }
    printf("\n");
}
int main()
{
    int n,i,val;
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    struct node*headbeg=NULL;
    for(i=0;i<n;i++)
    insertbeg(&headbeg,arr[i]);
    print(headbeg);
    struct node*headend=NULL;
    for(i=0;i<n;i++)
        insertend(&headend,arr[i]);
    print(headend);
    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.

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*next=NULL;
   struct node*prev=NULL;
};
void insertend(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;
```

```c
            }
    temp->next=newnode;
    newnode->prev=temp;
}
void print(struct node*head)
{
    struct node*temp=head;
    if(temp==0)
    {
        printf("List is empty");
    }
    while(temp!=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->next;
    }
    printf("\n");
}
int main()
{
    int n,i,val;
    struct node*head=NULL;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&val);
        insertend(&head,val);
    }
    print(head);
}
```

Status : Correct   *Marks : 10/10*