



Symbiosis Institute of Technology

Faculty of Engineering

CSE- Academic Year 2024-25

Data Structures – Lab Batch 2023-27

Lab Assignment No:- 4	
Name of Student	Pankhuri Varshney
PRN No.	23070122160
Batch	2023-27
Class	CS-B2
Academic Year & Semester	2024-25 Semester 3
Date of Performance	16 th August 2024
Title of Assignment:	<ol style="list-style-type: none">1. Creation of Singly Linked list for integers, Characters and Strings2. Count Number of Nodes3. Find the middle of the linked list. If the number of nodes is even, then there would be two middle nodes, so return the second middle node.4. Searching element in Singly Linked list5. Reverse the linked list by changing the links between nodes
Source Code/Algorithm/Flow Chart:	<p>Implement a program to:</p> <ol style="list-style-type: none">1. Create Singly Linked List2. Count Number of Nodes3. Find the Middle Node(s)4. Search element in Linked List <p>SOURCE CODE:</p> <pre>#include <stdio.h> #include <stdlib.h></pre>

```

struct node {
    int a;
    struct node* ptr;
};

void addNode(struct node **head, int nextData);
void printList(struct node* head);
int searchNode(struct node **head, int value);

void addNode(struct node **head, int nextData){
    struct node *newNode=(struct node*)malloc(sizeof(struct node));
    newNode->a=nextData;
    newNode->ptr=NULL;

    if(*head==NULL){
        *head=newNode;
    }
    else {
        struct node *temp=*head;
        while(temp->ptr!=NULL){
            temp=temp->ptr;
        }
        temp->ptr=newNode;
    }
}

void printList(struct node* head){
    struct node *temp=head;
    if(head==NULL)
        printf("LINKED LIST EMPTY\n");
    int i=1;
    while(temp!=NULL){
        printf("%d\t",temp->a);
        temp=temp->ptr;
    }
    printf("\n");
}

int countNodes(struct node* head){
    struct node *temp=head;
    int count=0;
    if(head!=NULL){
        while(temp!=NULL){
            count++;
            temp=temp->ptr;
        }
    }
    return count;
}

void middleNode(struct node *head){

```

```

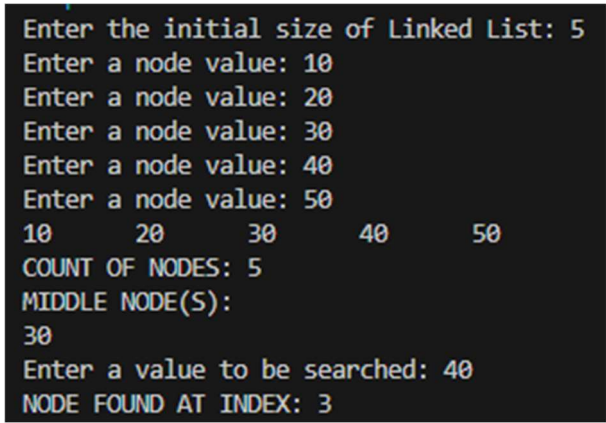
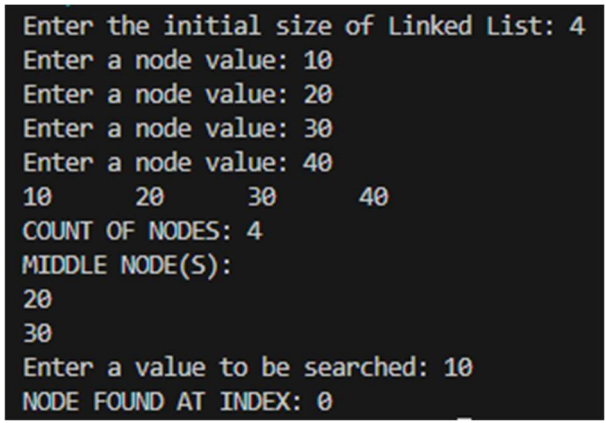
int count=countNodes(head);
struct node *temp=head;
printf("MIDDLE NODE(S):\n");
if (count%2!=0){
    for(int i=0;i<count/2; i++){
        temp=temp->ptr;
    }
    printf("%d\n", temp->a);
}

if(count%2==0){
    for(int i=0;i<count/2-1; i++){
        temp=temp->ptr;
    }
    printf("%d\n", temp->a);
    temp=temp->ptr;
    printf("%d\n", temp->a);
}
}

int searchNode(struct node **head, int value){
    if(*head==NULL){
        return -1;
    }
    else{
        struct node *temp=*head;
        int i=0;
        while(temp->ptr!=NULL&&temp->a!=value){
            temp=temp->ptr;
            i++;
        }
        if(temp->a==value){
            return i;
        }
        else{
            return -1;
        }
    }
}

int main(){
    struct node* head=NULL;
    printf("Enter the initial size of Linked List: ");
    int size;
    scanf("%d", &size);
    for(int i=0; i<size; i++){
        printf("Enter a node value: ");
        int val;
        scanf("%d", &val);
        addNode(&head, val);
    }
    printList(head);
}

```

	<pre> printf("COUNT OF NODES: %d\n", countNodes(head)); middleNode(head); int value; printf("Enter a value to be searched: "); scanf("%d", &value); int index=searchNode(&head, value); if(index==-1){ printf("NODE NOT PRESENT\n"); } else printf("NODE FOUND AT INDEX: %d\n", index); return 0; } </pre>
Output Screenshots	<p>CASE 1: Odd number of nodes in the list</p>  <pre> Enter the initial size of Linked List: 5 Enter a node value: 10 Enter a node value: 20 Enter a node value: 30 Enter a node value: 40 Enter a node value: 50 10 20 30 40 50 COUNT OF NODES: 5 MIDDLE NODE(S): 30 Enter a value to be searched: 40 NODE FOUND AT INDEX: 3 </pre> <p>CASE 2: Even number of nodes in the list</p>  <pre> Enter the initial size of Linked List: 4 Enter a node value: 10 Enter a node value: 20 Enter a node value: 30 Enter a node value: 40 10 20 30 40 COUNT OF NODES: 4 MIDDLE NODE(S): 20 30 Enter a value to be searched: 10 NODE FOUND AT INDEX: 0 </pre>
Practice questions	<ol style="list-style-type: none"> 1. Create a structure called "Student" with members name, age, and total marks. Write a C program to input data for two students, display their information, and find the average of total marks. 2. Define a structure named Time with members hours, minutes, and seconds. Write a C program to input two times, add them, and display the result in proper time format.

	<ol style="list-style-type: none">3. Create a structure named Book to store book details like title, author, and price. Write a C program to input details for three books, find the most expensive and the lowest priced books, and display their information.4. Write a program in C to add numbers using call by reference.5. Write a program in C to find the maximum number between two numbers using a pointer.
Conclusion	Thus, we have studied the concept of Linked List and how it is different from arrays.