

Subject: - DSU	Subject Code: 313 301
Semester: - III	Course: Computer Engineering
Laboratory No: L003	Name of Subject Teacher: Prof. Imraan S.
Name of Student: SAAD SHARIF KAZI	Roll Id: - 24203A0013
Experiment No:	12
Title of Experiment	* Write a 'C' Program to Implement Singly Linked List with Operations: (i) Insert at beginning, (ii) Search, (iii) Display

Aim: Write a 'C' Program to Implement Singly Linked List with Operations: (i) Insert at beginning, (ii) Search, (iii) Display

Algorithm:

Algorithm for Linked List Program

Step 1: Start

Step 2: Define a structure Node with two members:

- a) data (integer)
- b) next (pointer to next node)

Step 3: Declare functions:

- createlinkedlist(n) – to create a linked list of n nodes
- createnode(data) – to create a new node
- printList(head) – to display the linked list
- begin(head) – to insert a node at the beginning
- search(head) – to search an element in the linked list

Step 4: In main():

- a) Read number of nodes n
- b) Call createlinkedlist(n) and store the returned head pointer
- c) Call printList(head) to display the list
- d) Call begin(head) to insert a new node at the beginning and update head
- e) Call printList(head) again to show updated list
- f) Call search(head) to find an element

Function: createlinkedlist(n)

Step 1: If $n \leq 0$, print error and return NULL

Step 2: Read data for first node

- a) Call createnode(data) to create the node
- b) Assign it to head and temp

Step 3: Repeat from i = 2 to n

- a) Read data for next node
- b) Create new node using createnode(data)
- c) Link temp->next = newNode

d) Move temp = newNode

Step 4: Return head pointer

Function: createnode(data)

Step 1: Allocate memory dynamically for a new node

Step 2: Assign newNode->data = data

Step 3: Set newNode->next = NULL

Step 4: Return pointer to newNode

Function: printList(head)

Step 1: Set temp = head

Step 2: While temp != NULL

a) Print temp->data

b) Move to next node temp = temp->next

Step 3: Print "NULL" at the end

Function: begin(head)

Step 1: Read data for new node

Step 2: Create new node using createnode(data)

Step 3: Point newNode->next = head

Step 4: Return newNode as new head

Function: search(head)

Step 1: Read element data to search

Step 2: Initialize counter = 1, temp = head

Step 3: Traverse while temp != NULL

a) If temp->data == data, print that data is found at position counter

b) Otherwise move to next node and increment counter

Step 4: If loop ends without match, print "Element not found"

CODE:

```
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SAAD12.C 1=[+]
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

struct Node
{
int data;
struct Node* next;
};

struct Node* createlinkedlist(int);
struct Node* createnode(int);
void printList(struct Node*);
struct Node* begin(struct Node*);
void search(struct Node*);

void main()
{
int n;
struct Node* head=NULL;
clrscr();
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SAAD12.C 1=[+]
clrscr();
printf("Enter the No. of Nodes: ");
scanf("%d",&n);
head=createlinkedlist(n);
printList(head);
head=begin(head);
printList(head);
search(head);
getch();
}

struct Node* createlinkedlist(int n)
{
int data,i;
struct Node *head=NULL,*temp=NULL,*newNode=NULL;
if(n<=0)
{
printf("Number of Nodes should be greater than Zero...");
return NULL;
}
printf("Enter data for Node 1: ");
41:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
```

```
File Edit Search Run Compile Debug Project Options Window Help
SAAD12.C 1=1
printf("Enter data for Node 1: ");
scanf("%i",&data);
newNode=createnode(data);
head=newNode;
temp=newNode;
for(i=2;i<=n;i++)
{
printf("Enter data for Node %i: ",i);
scanf("%i",&data);
newNode=createnode(data);
temp->next=newNode;
temp=newNode;
}
return head;
}

struct Node* createnode(int data)
{
struct Node* newNode=(struct Node*)malloc(sizeof(struct Node));
if(!newNode)
{
* 62:24
```

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
File Edit Search Run Compile Debug Project Options Window Help
SAAD12.C 1=1
if(!newNode)
{
printf("Memory Allocation Error...");
return NULL;
}
newNode->data=data;
newNode->next=NULL;
return newNode;
}

void printList(struct Node* head)
{
struct Node* temp=head;
while(temp!=NULL)
{
printf("%i -> ",temp->data);
temp=temp->next;
}
printf("NULL\n");
}
* 81:24
```

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
File Edit Search Run Compile Debug Project Options Window Help
SAAD12.C 1=[↑]

struct Node* begin(struct Node* start)
{
    int data;
    struct Node *temp=start,*newNode=NULL;
    printf("\nEnter data for Node to be inserted in the Beginning: ");
    scanf("%i",&data);
    newNode=createnode(data);
    newNode->next=temp;
    return newNode;
}

void search(struct Node* head)
{
    int data,count=1;
    struct Node* temp=head;
    printf("\nEnter data you want to search in the linked list: ");
    scanf("%i",&data);
    while(temp!=NULL && temp->data!=data)
    {
        temp=temp->next;
        * 101:24

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
while(temp!=NULL && temp->data!=data)
{
    temp=temp->next;
    count++;
}
if(temp!=NULL)
    printf("%i found in Node %i",data,count);
else
    printf("%i not found in the list",data);
}

* 109:24

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

OUTPUT: -

```
Enter the No. of Nodes: 3
Enter data for Node 1: 12
Enter data for Node 2: 34
Enter data for Node 3: 45
12 -> 34 -> 45 -> NULL
```

```
Enter data for Node to be inserted in the Beginning: 2
2 -> 12 -> 34 -> 45 -> NULL
```

```
Enter data you want to search in the linked list: 34
34 found in Node 3_
```

Practical Related Questions:

1. Write a function to insert a node at the beginning of a Singly Linked List.

Ans:

```
struct Node* begin(struct Node* start)
{
    int data;
    struct Node *temp = start, *newNode = NULL;
    printf("\nEnter data for Node to be inserted in the Beginning: ");
    scanf("%i",&data);
    newNode = createnode(data);
    newNode->next = temp;
    return newNode;
}
```

2. Write a function to detect if a singly linked list has a cycle. If a cycle is detected, return the starting node of the cycle.

Ans:

```
void cyclecheck(struct Node* head)
{
    int flag = 0;
    struct Node *temp = NULL;
    temp=head->next;
    while(temp!=NULL)
    {
        if(temp==head)
        {
            printf("\nThere is a Cycle, Address of the first node of the cycle is %u",temp);
            flag = 1;
            break;
        }
        temp=temp->next;
    }
    if(flag==0)
    {
        printf("There is not cycle!");
    }
}
```

Marks Obtained	Dated signature of Teacher
-----------------------	-----------------------------------

Process Related (35)	Product Related (15)	Total (50)	